(19)日本国特許庁（ＪＰ）　(12)**公開特許公報（Ａ）**　(11)特許出願公開番号
特開2002－366029
（P2002－366029A）
(43)公開日　平成14年12月20日(2002.12.20)

(51)Int.Cl.[7]	識別記号	ＦＩ	テーマコード*（参考）
Ｇ０９Ｃ　1/00	６１０ ６５０	Ｇ０９Ｃ　1/00	６１０Ｂ　５Ｊ１０４ ６５０Ｂ

審査請求　未請求　請求項の数６　ＯＬ　（全 28 頁）

(21)出願番号　特願2001－178407(P2001－178407)

(22)出願日　平成13年６月13日(2001.6.13)

(71)出願人　000005223
富士通株式会社
神奈川県川崎市中原区上小田中４丁目１番１号

(72)発明者　伊藤　孝一
神奈川県川崎市中原区上小田中４丁目１番１号　富士通株式会社内

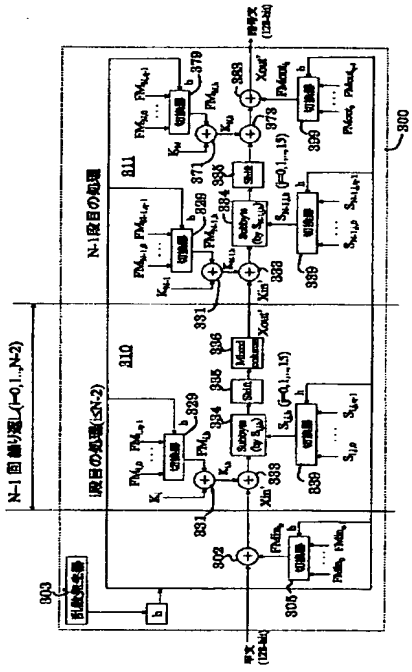
(72)発明者　武仲　正彦
神奈川県川崎市中原区上小田中４丁目１番１号　富士通株式会社内

(74)代理人　100062993
弁理士　田中　浩　（外２名）

最終頁に続く

(54)【発明の名称】　ＤＰＡに対して安全な暗号化

(57)【要約】
【課題】　固定値を用いてマスクを行うことによって処理速度の向上と必要なＲＡＭ領域の削減を実現する。
【解決手段】　暗号化装置（３００）は、乱数を発生する乱数発生器手段と、乱数に従ってｑ個の固定値の中の１つを選択する第１の選択器（３２９）と、乱数に従ってｑ組の固定テーブルの中の１組を選択する選択器（３３９）と、を具えている。排他的論理和手段（３３３）は、固定値と鍵の排他的論理和と入力の排他的論理和をとる。非線形変換手段（３３４）は、１組の固定テーブルに従って非線形変換を行う。別の暗号化装置（５００）は、並列に結合された複数の暗号化部（５１１、５１３）と、乱数に従ってその複数の暗号化部の中の１つを選択する選択器（５０２）と、を具えている。



1

【特許請求の範囲】

【請求項１】　排他的論理和手段および非線形変換手段を有する暗号化装置であって、

乱数を発生する乱数発生手段と、ｑ個（ｑは整数）の固定値と、前記乱数に従って前記ｑ個の固定値の中の１つを選択する第１の選択器と、を具え、

前記排他的論理和手段は、前記選択された固定値と鍵の排他的論理和と前記排他的論理和手段の入力の排他的論理和をとるものであること、を特徴とする、暗号化装置。

【請求項２】　排他的論理和手段および非線形変換手段を有する暗号化装置であって、

乱数を発生する乱数発生手段と、ｑ組（ｑは整数）のマスクされた固定テーブルと、前記乱数に従って前記ｑ組の固定テーブルの中の１組を選択する選択器と、を具え、

前記非線形変換手段は、前記選択された１組の固定テーブルに従って入力を非線形変換するものであること、を特徴とする、暗号化装置。

【請求項３】　乱数を発生する乱数発生手段と、並列に結合された複数の暗号化部と、前記乱数に従って前記複数の暗号化部の中の１つを選択する選択器と、を具え、

前記複数の暗号化部の各々は、排他的論理和手段および非線形変換手段を含むものであること、を特徴とする、暗号化装置。

【請求項４】　乱数を発生する乱数発生手段と複数の暗号化段とを具えた暗号化装置であって、

前記複数の暗号化段の各々は、入力を非線形変換する非線形変換手段と、第１の入力と第２の入力の排他的論理和をとる排他的論理和手段と、を含み、

前記排他的論理和手段の第２の入力は前記非線形変換手段の出力に結合されており、

前記非線形変換手段は、ｑ個（ｑは整数）の固定値と、前記乱数に従って前記ｑ個の固定値の中の１つを選択する選択器と、前記選択された固定値と鍵の排他的論理和と入力の排他的論理和をとる別の排他的論理和手段と、を有するものであること、を特徴とする、暗号化装置。

【請求項５】　乱数を発生する乱数発生手段と、並列に結合された複数の暗号化部と、前記乱数に従って前記複数の暗号化部の中の１つを選択する選択器と、を具え、

前記複数の暗号化部の各々は複数の暗号化段を有し、

前記複数の暗号化段の各々は、入力を非線形変換する非線形変換手段と、第１の入力と第２の入力の排他的論理和をとる排他的論理和手段と、を含み、

前記排他的論理和手段の第２の入力は前記非線形変換手段の出力に結合されているものであること、特徴とする、暗号化装置。

【請求項６】　暗号化装置において用いられるプログラムであって、

乱数に従って複数の暗号化手順の中の１つを選択するステップと、前記選択された暗号化手順に従って、入力値を暗号化して出力するステップと、を実行させ、

前記暗号化するステップは、固定値と鍵の排他的論理和と入力値の排他的論理和をとるステップと、１組の固定テーブルに従って入力値を非線形変換するステップと、を含むものであること、を特徴とする、プログラム。

【発明の詳細な説明】

【０００１】

【発明の属する技術分野】本発明は、情報の暗号化の分野に関し、特に電力解析攻撃と呼ばれる暗号解読法に対する暗号のセキュリティ（安全）のための技術に関する。

【０００２】

【発明の背景】暗号方式には一般的に公開鍵暗号方式と共通鍵暗号方式が含まれる。いわゆる共通鍵暗号方式は、暗号化と復号で同一の秘密鍵（キー）を用いる。この秘密鍵を送信機側のユーザと受信機側のユーザの間で共有しその他の者に対して秘密にすることによって、暗号文が安全に送信できる。図１はスマートカードにおける共通秘密鍵を用いた暗号化の例を示している。図１において、スマートカードは、周知の形態で入力の平文（プレインテキスト）をその内部の暗号化処理部において共通秘密鍵を用いて暗号化して出力の暗号文を供給する。

【０００３】暗号解読は、秘密鍵を含めた秘密情報を、暗号文等入手可能な情報から推定する。暗号解読の１つである電力解析攻撃が、Paul Kocher, Joshua Jaffe, and Benjamin Jun, “Differential Power Analysis” in proceedings of Advances inCryptology－CRYPTO’９９, Springer-Verlag, １９９９, pp. ３８８－３９７に記載されている。この電力解析攻撃は、スマートカード等に搭載された暗号化プロセッサに様々な入力データを与えた時の電力消費データを収集および解析して、暗号プロセッサ内部の鍵情報を推定する。これは、公開鍵暗号と秘密鍵暗号の双方に適用できる。

【０００４】電力解析攻撃には、単純電力解析（以下、ＳＰＡという）（Single Power Analysis）および電力差分攻撃（以下、ＤＰＡという）（Differential Power Analysis）が含まれる。ＳＰＡは暗号プロセッサにおける単一の電力消費データの特徴から秘密鍵の推定を行う。ＤＰＡは相異なる多数の電力消費データの差分を解析することによって秘密鍵の推定を行い、一般的にはＤＰＡの方が強力である。

【０００５】例えばＲＳＡ等の公開鍵暗号に対するＤＰＡが、例えば、Thomas S. Messerges, Ezzy A. Dabbish and Robert H. Sloan “Power Analysis Attacks of Modular Exponentitiation in Smartcards” Cryptographic Hardware and Embedded Systems (CHES’９９), Springer-Verlag, pp. １４４－１５７に記載されている。共通鍵暗号方式の現在の標準のＤＥＳ（Data Encryptio

3

n Standards）に対するＳＰＡおよびＤＰＡが、Paul Kocher, Joshua Jaffe, and Benjamin Jun, " Differential Power Analysis, " in proceedings of Advances in Cryptology－CRYPTO' ９９, Springer-Verlag, １９９９, pp. ３８８－３９７に記載されている。共通鍵暗号方式の次世代標準となりうるラインデール（Rijndael）法に対するＤＰＡが、例えば、S. Chari, C. Jutla, J. R. Rao, P. Rohatgi, "An Cautionary Note Regarding Evaluation of AES Candidates on Smart-Cards," Second Advanced Encryption Standard Candidate Conference, March１９９９に記載されている。

【０００６】このように、電力解析攻撃の中でもＤＰＡは特に有効な方法として注目されており、様々なＤＰＡ暗号解読法が研究されている。一方、ＤＰＡ暗号解読を防止するための技術も開発されている。

【０００７】ＤＰＡを適用可能な共通鍵暗号の通常の典型的な構成を説明する。図２、３および４は、典型的な共通鍵暗号で用いられる処理（operations）である鍵ＸＯＲ（排他的論理和）、線形変換および非線形変換をそれぞれ示している。

【０００８】図２において、鍵ＸＯＲは入力データＸｉの鍵情報Ｋｉとの排他的論理和（ＸＯＲ）の出力Ｚｉを供給する。（但し、演算ＸＯＲは図面および式では、○と＋を合成したシンボルで示されている。）図３において、線形変換Ｌは入力データＸｉに対する線形変換された出力Ｚｉ＝Ｌ（Ｘｉ）を供給する。ここで、任意のｘおよびｙについてＬ（ｘＸＯＲｙ）＝Ｌ（ｘ）ＸＯＲＬ（ｙ）である。線形変換には、ビット転置（permutation）および行列演算等がある。図４において、非線形変換Ｗは、入力データＸｉを非線形変換して出力Ｚｉ＝Ｗ（Ｘｉ）を供給する。ここで、任意のｘよびｙについてＷ（ｘＸＯＲｙ）≠Ｗ（ｘ）ＸＯＲＷ（ｙ）である。非線形変換は、典型的には非線型変換テーブルＳｂｏｘをしばしば用い、入力ＸをＸ＝｛ｘｕ－１，．．．，ｘ１，ｘ０｝（ｕは自然数）とｕ個に分割し、Ｓｂｏｘであるｗｉ（ｉ＝０～ｕ）を用いてｚｉ＝ｗｉ（ｘｉ）を演算し、Ｚ＝（ｚｕ－１．．．ｚ１ｚ０）として結合＊して出力Ｚを生成する。

4

【０００９】典型的な共通鍵暗号では、これらの鍵ＸＯＲ演算、線形変換および非線形変換の３つを適当に組み合わせて各ラウンド関数を構成し、そのようなラウンド関数を順に複数回（ラウンド）繰り返す。

【００１０】ＤＰＡによる暗号解読技術を説明する。ＤＰＡは、電力消費データの測定の段階と、その電力消費データの差分に基づく鍵の解析の段階とを含んでいる。電力消費データの測定において、相異なる複数の符号のシーケンスを含んだ入力平文をスマートカード等の暗号化器にシリアルに供給して、その暗号化プロセッサによってその入力平文に応答して消費される電力消費の時間変化を例えばオシロスコープ等を用いて測定して、電力消費曲線を得る。図７（Ａ）はそのような電力消費曲線を例示している。相異なる入力平文に対してその測定を行って統計的に充分な数の電力消費曲線を収集する。その測定で得られた複数の電力消費曲線の集合（セット）をＧとする。

【００１１】次に電力消費曲線を用いた鍵の解析を説明する。図５は、従属接続の形の鍵ＸＯＲ演算（図２）と非線形変換（図４）の組合せである暗号化の例を示している。この暗号化に対するＤＰＡを説明する。図６は、図５における任意の非線形変換要素ｗｉに関係する構成要素を取り出したものである。図６において、任意の入力平文における既知の数ビット値をｍｉとし、未知の鍵Ｋにおける要素値をｋｉ（Ｋ＝｛ｋｕ－１，．．．，ｋ１，ｋ０｝の中の１つの要素）とし、既知のＳｂｏｘテーブルにおける変換関数をｗｉとし、出力をｚｉ（＝ｗｉ（ｍｉＸＯＲｋｉ））とする。ＤＰＡのために、プロセッサ内部で用いられている鍵の要素値を任意のｋｉ'であると仮定する。既知のｍｉおよびｗｉとその仮定されたｋｉ'とからｚｉ'＝ｗｉ（ｍｉＸＯＲｋｉ'）を演算して、仮定されたｋｉ'に対する集合Ｇ（ｋｉ'）を次の部分集合Ｇ０（ｋｉ'）およびＧ１（ｋｉ'）に分ける。

【００１２】

【数１】

$$G_0(k'_i) = \{ G \mid z'_i = w_i(m_i \oplus k'_i) \text{の第e桁のビット値}=0 \} \quad (1)$$

【数２】

$$G_1(k'_i) = \{ G \mid z'_i = w_i(m_i \oplus k'_i) \text{の第e桁のビット値}=1 \} \quad (2)$$

ここで、ｅは、自然数であり、ｅ番目のＬＳＢを表している。

【００１３】次いで、仮定されたｋｉ'に対する次の電※力消費曲線の差分ＤＧ（ｋｉ'）を作成する。

【数３】

$$DG(k'_i) = (G_1\text{に属する電力消費曲線の平均}) - (G_0\text{に属する電力消費曲線の平均}) \quad (3)$$

【００１４】図７（Ａ）はＧ１に属する電力消費曲線の平均電力消費曲線を例示している。図７（Ｂ）はＧ０に属する電力消費曲線の平均電力消費曲線を例示している。仮定された鍵の要素値が真の鍵の要素値と等しく、即ちｋｉ'＝ｋｉである場合には、図７（Ａ）と（Ｂ）の差を表す図７（Ｃ）に例示された差分電力消費曲線中にスパイクが現われる。仮定された鍵の要素値が真の鍵の要素値と等しくなく、即ちｋｉ'≠ｋｉである場合に

5

は、図７（Ａ）と（Ｂ）の差を表す図７（Ｄ）に例示された差分電力消費曲線は概ね平坦な曲線となる。従って、仮定したｋｉ’に基づいて生成されたスパイクを有する差分電力消費曲線から、鍵ｋｉが推定できる。全てのｉに対するｋｉについて差分電力消費曲線を生成することによって、最終的に鍵Ｋを解読または確定（determine）することができる。

【００１５】次に、ｋｉ’＝ｋｉである場合に、電力差分曲線ＤＧ（ｋｉ’）にスパイクが現れることを説明する。ｋｉ’＝ｋｉである場合、集合Ｇ（ｋｉ’）を式（１）および式（２）に従って部分集合Ｇ０（ｋｉ’）＊

6

＊およびＧ１（ｋｉ’）に分けると、仮定したｚｉ’＝ｗｉ（ｍｉＸＯＲｋｉ’）とこれに対応するプロセッサ内における実際のｚｉ＝ｗｉ（ｍｉＸＯＲｋｉ）とが全てのｍｉに対して一致するので、ｚｉのハミング・ウェイトＨＷ（Hamming weight）に関して次の式（４）が得られる。ハミング・ウェイトとは或る値をビット値で表現したときのビット値＝１の個数である。例えば２進４ビット値（１１０１）２のハミングウェイトＨＷは３である。

【数４】

$$(G_1\text{に属する}z_i\text{の平均HW}) - (G_0\text{に属する}z_i\text{の平均HW}) = 1 \qquad (4)$$

【００１６】一方、ｋｉ’≠ｋｉの場合は、仮定されたｚｉ’＝ｗｉ（ｍｉＸＯＲｋｉ’）とそれに対応するプロセッサ内における実際のそれぞれのｚｉ＝ｗｉ（ｍｉＸＯＲｋｉ）とが無関係なので、仮定されたｚｉ’に対する式（１）および式（２）に従って全てのｍｉに対する集合Ｇ（ｋｉ’）を部分集合Ｇ０（ｋｉ’）およびＧ＊１（ｋｉ’）に分けても、実際のそれぞれのｚｉ（即ちｚｉ’と仮定された実際のｚｉ）に対しては２つの部分集合にランダムに分けたことになるので、次の式（５）が成立する。

【数５】

$$(G_1\text{に属する}z_i\text{の平均HW}) - (G_0\text{に属する}z_i\text{の平均HW}) \sim 0 \qquad (5)$$

【００１７】式（４）が成立する場合はＧ０（ｋｉ’）とＧ１（ｋｉ’）の間でのロード値ｚｉの平均ハミング・ウェイトに大きな差が生じるが、式（５）が成立する場合はＧ０（ｋｉ’）とＧ１（ｋ’）の間でのロード値ｚｉの平均ハミングウェイトに大きな差が生じない。

【００１８】ｚｉ＝ｗｉ（ｘｉ）で示される変換ｗｉは、暗号化器内のＲＯＭやＲＡＭなどのメモリ上から変換テーブルＳｂｏｘの出力値ｚｉがロード命令によって読み込まれることによって行われる。ロード命令が実行されるとき、ロード値のハミングウェイトに比例した電力が消費されるものと考えられている。その考えの妥当性を示す実験結果が、T. S. Messerge, Ezzy A. Dabbish and Robert H. Sloan, “Investigations of Power Attacks on Smartcards” Proceedings of USENIX Workshop on Smartcard Technology, Mar. １９９９に記載されている。

【００１９】従って、ｋｉ’＝ｋｉである場合、式（４）が満たされることで消費電力の差が電力差分曲線上のスパイクという形になって現れるが、式（５）の場合はスパイクが現れず、概ね平坦な曲線となる。ＤＰＡは、図４に図３の線形変換Ｌを挿入した暗号化器にも適用できることが判明している。

【００２０】図８は、図４の暗号化器の前後に2つの線★形変換を追加した構成を有する暗号化器を示している。L_1およびL_2をビット転置関数とし、ｗｉをＤＥＳのＳｂｏｘとすると、図８の構成はＤＥＳのＦ関数と等価になる。ＤＥＳの仕様については、FIPS ４６, “Data encryption standard” Federal Information ProcessingStandards Publication ４６, U.S. Department of Commerce/ National Bureau of Standards, National Technical Information Service, Springfield, Virginia, １９７７を参照されたい。図８の処理も、図６と同様の処理に変換でき、同様にＤＰＡを適用して鍵Ｋを推定できる。

【００２１】以上の手法では、非線形変換中のＳｂｏｘ出力に注目してＤＰＡを適用している。さらに、入力ｍｉと鍵ｋｉのＸＯＲの値（鍵ＸＯＲ演算の出力）やＳｂｏｘへの入力値ｘｉに対してＤＰＡを適用する手法もある。また、或るプロセッサにおいては、隣接ビットモデルによって次の式（６）で表される電力消費がロード値のビットの関数で表現でき、それによってより有効な解析が可能になる場合がM. Akkar, R. Bevan, P. Dischamp, and D. Moyart, “Power Analysis, What IsNow Possible...” Asiacrypt ２０００において報告されている。

【数６】

$$V(z) = a' + a_0z_0 + a_1z_1 + \ldots + a_7z_7 + a_{0,1}z_0z_1 + a_{1,2}z_1z_2 + \ldots + a_{6,7}z_6z_7 \qquad (6)$$

【００２２】これらの手法によれば、次の場合にＤＰＡによって秘密鍵Ｋが解読される。図９は、図４の暗号化器における電力消費曲線の測定対象点Ａ、ＢおよびＣを示している。

【００２３】１．入力Ｍが既知かつ攻撃者が自由に選択可能即ち制御可能であり、鍵Ｋが未知の固定値であり、Ｓｂｏｘのｗｉの変換が既知の場合。この場合、図９の測定対象点Ａの所定タイミングにおける（Ｓｂｏｘｗｉの出力の）電力消費曲線が測定される。

【００２４】２．入力Ｍが既知かつ制御可能であり、鍵Ｋが未知の固定値である場合。この場合、図９の測定対象点Ｂの所定タイミングにおける（鍵ＸＯＲ演算の出力

の）電力消費曲線が測定される。

【００２５】３．入力Mが既知かつ制御可能であり、鍵Kが未知の固定値である場合。この場合、図９の測定対象点Cの所定タイミングにおける（Ｓｂｏｘのw_iをインデックスするためのロード入力の）電力消費曲線が測定される。

【００２６】従来のＤＰＡ対策

従来のＤＰＡ対策法としては、例えばスマートカード内にノイズ発生器を置くことによって電力消費量の測定精度を落とす手法と、暗号アルゴリズムに対策を施す手法が存在する。その測定精度を落とす手法は容易に行えるが、測定回数を増加するなどの方法で解析が可能になるので根本的な対策とはならない。それに対して、暗号アルゴリズムへの対策は、容易でないが根本的な対策となりうる。暗号アルゴリズムに対策を施す手法の代表的なものとして、マスキング法と呼ばれているThomas S. Messerges, "Securing the AES Finalists Against Power Analysis Attacks," in proceedings of Fast Software Encryption Workshop ２０００, Springer-Verlag, April ２０００が知られている。マスキング法とは、入力値Mそのものを用いて暗号化の各処理を行う代わりに、乱数Ｒをマスクとして$M' = MXORR$で表される値M'を用いて暗号化を行う方法である。乱数Ｒは暗号化の各処理ごとに発生させるので、ここでは"乱数マスク値法"と呼ぶ。

【００２７】次に、乱数マスク値法について説明する。図１０は、乱数マスク値法による処理の概要図を示している。この処理は、図示のような上側の暗号化処理部と下側のマスク値生成部と乱数発生とで構成される。

【００２８】図２、３および４に示されている通常の鍵ＸＯＲ関数、線形関数および非線形関数が用いられる通常の暗号化処理を図１０の暗号化処理に変更する場合は、それらは図１０の暗号化において図１１、１２および１３に示されているような乱数マスク値法による鍵ＸＯＲ関数、線形関数および非線形関数に置換される。

【００２９】乱数マスク値法では、暗号化における通常の中間データX_iを計算する代わりに排他的論理和$X_i = X_i' XORR_i$を満たすX_i'および乱数R_iが計算される。暗号化処理部においてX_i'が計算され、マスク値生成処理部においてR_iが計算される。図２と図１１、図３と図１２、および図４と図１３に示されている処理（operation）におけるX_i、X_i'、Z_i、Z_i'、R_iおよびRO_iに関して、次の式（７）が成立する。

【数７】

$$\begin{cases} Xi = Xi' \oplus Ri \\ Zi = Zi' \oplus ROi \end{cases} \quad (7)$$

【００３０】図２では入力値X_iと鍵K_iに対して$Z_i = X_i XORK_i$が演算されるのに対して、図１１では

暗号化処理内部で乱数発生器によって乱数RK_iを発生させた後、入力値X_i'と鍵K_iに対して$Z_i' = X_i' XORK_i XORRK_i$が演算される。また、マスク値生成において$R_i$と$RK_i$から$RO_i = R_i XORRK_i$が演算される。

【００３１】図３では$Z_i = L(X_i)$の線形変換が行われるのに対して、図１２の暗号化処理では$Z_i' = L(X_i')$の変換が行われ、マスク値生成では$RO_i = L(R_i)$の変換が行われる。

【００３２】図４ではw_1、w_2、・・・、w_{u-1}で表されるｕ個のＳｂｏｘを用いた非線型変換が行われるのに対して、図１３（Ａ）の暗号化処理では、図のＮｅｗＳｂｏｘを用いた処理によってＲＡＭ領域上にwi'_1、wi'_2、・・・、wi'_{u-1}で表される新しいＳｂｏｘが作成されて、これを用いて非線型変換が行われる。また、図１３（Ａ）のマスク値生成ではそのＮｅｗＳｂｏｘを用いた処理が行われ、R_iと内部で生成された乱数RO_iからw'_1、w'_2、・・・、w'_{u-1}が生成され、w'_1、w'_2、・・・、w'_{u-1}とRO_iが出力される。図１３（Ｂ）はＮｅｗＳｂｏｘの詳細構成を示している。ＮｅｗＳｂｏｘは、内部の乱数発生器を用いてRO_iを生成し、また、$R_i = r_{iu-1} \cdots r_{i1} r_{i0}$と、$RO_i = ro_{iu-1} \cdots ro_{i1} ro_{i0}$と、図１２（Ｂ）内で用いられるＳｂｏｘ、$w_1$、$w_2$、・・・、$w_{u-1}$とから、$wi'_j(x) = w(x XOR r_{ij}) XOR ro_{ij}$を満たす$wi'_j$をｊ＝０、１、・・・、ｕ－１について作成し、そしてそのRO_iとwi'_jを出力する。

【００３３】次に、乱数マスク値法の安全性について簡単に説明する。乱数マスク値法においては、図１０および後で説明する図１９中の各ラウンド（段）における図１３（Ａ）および（Ｂ）のＳｂｏｘのwi'が乱数に従って変化するので、ＤＰＡにおいてＳｂｏｘの内容を知ることができない。即ち、前述の条件１におけるＳｂｏｘが既知であるという条件が成立しないので、図８の測定対象点Ａの所定のタイミングにおいて測定された電力消費曲線を式（１）および（２）に従ってG_0とG_1に分けることができない。従って、乱数値マスク法を用いた暗号処理装置はＤＰＡに対して安全である。同様に、前述の条件２および３における鍵ＸＯＲ関数の出力における測定対象点ＢおよびＳｂｏｘへの入力における測定対象点Ｃについても、測定毎に変化する乱数要素が加算されるので、鍵Kが固定であるという条件が成立しないので、ＤＰＡに対して安全である。

【００３４】次に、乱数マスク値法を用いた暗号化の例として、ラインデール法を説明する。図１４は、ＤＰＡ対策のない通常のＮ段ラインデール処理の全体的構成を示している。Ｎ段ラインデール処理の各段は、ＸＯＲ演算、サブバイト処理Ｓｕｂｂｙｔｅ（Substitute Byt

e）、シフト処理Ｓｈｉｆｔ、およびミクストコラム処理Ｍｉｘｅｄｃｏｌｕｍｎを含んでいる。但し、最後の段は、別のＸＯＲ演算を含んでおり、ミクストコラム処理を含んでいない。図１４において、Ｎは、秘密鍵Ｋｓｅｃのビット数によって決定され、Ｋｓｅｃが１２８ビットの場合はＮ＝１０とし、１９２ビットの場合はＮ＝１２とし、および２５６ビットの場合はＮ＝１４とする。K_iは拡大鍵（sub-key）と呼ばれる鍵である。図１５は、ラインデール法の１２８／１９２／２５６ビット秘密鍵ＫｓｅｃからＮ＋１個の１２８ビット拡大鍵K_0、K_1、・・・、K_Nを生成する拡大鍵生成器を示している。秘密鍵から拡大鍵を生成する方法は、http://www.nist.gov/aes/に示されているRijndaelの仕様書に記載されている。

【００３５】図１６はサブバイト処理Ｓｕｂｂｙｔｅを示しており、この処理は、８ビット－８ビット変換ＳｂｏｘであるＳを用いて、１２８ビット－１２８ビット非線形変換を行う。図１７はシフト処理Ｓｈｉｆｔを示しており、この処理は、バイト単位の並べ替えを行う処理である。図１８はミクストコラム処理Ｍｉｘｅｄｃｏｌｕｍｎを示しており、この処理は、行列を用いた体ＧＦ（２８）上の演算を行う。

【００３６】図１９は、図１４の通常のＮ段ラインデール法に対して、乱数マスク値法を適用したＮ段ラインデール法を示している。図１９のＮ段ラインデール法は、図示のような上側のＮ段の暗号化処理部と下側のＮ段のマスク値生成部と乱数発生とで構成される。K_iはラインデール法の第ｉ段の拡大鍵を表す。RK_iはそれぞれの拡大鍵に対するランダムなマスク値を表す。サブバイト処理Ｓｕｂｂｙｔｅは、図１６に示されているような形態で１６個のＳｂｏｘ：$S_{i,0}$、$S_{i,1}$、・・・、$S_{i,15}$を用いて１２８ビット－１２８ビット非線形変換を行う。$S_{i,0}$、$S_{i,1}$、・・・、$S_{i,15}$は第ｉ段の新しいＳｂｏｘ“ＮｅｗＳｂｏｘ”によって生成されたＳｂｏｘである。図２０は、ＮｅｗＳｂｏｘを示しており、入力値Rin_iから、内部で発生した乱数$Rout_i$に従って相異なる１６個のＳｂｏｘ、$S_{i,0}(x)$、$S_{i,1}(x)$、・・・、$S_{i,15}(x)$を生成し、乱数$Rout_i$を供給する。Ｓｈｉｆｔ処理およびＭｉｘｅｄｃｏｌｕｍｎ処理は、通常のラインデール法の処理と同様に、図１７および図１８に示す線形変換である。

【００３７】図１９の処理のフローをステップ［１１０１］～［１１０９］およびステップ［１２０１］～［１２０９］で示す。

［１１０１］　ｉ＝０とセットする。

［１１０２］　乱数マスク値Ｒｉｎを生成し、平文に対してＲｉｎとのＸＯＲ演算を行う（ＸＯＲする）。

［１１０３］　平文に対してK_iＸＯＲRK_iとのＸＯＲ演算を行う。マスク処理にマスク値RK_iを入力して１６個のＳｂｏｘ、$S_{i,j}(x)$（ｊ＝０、１、・・・、１５）を生成する。

［１１０４］　ステップ［１１０３］で生成した$S_{i,j}(x)$を用いたＳｕｂｂｙｔｅ処理を行う。

［１１０５］　Ｓｈｉｆｔ処理およびＭｉｘｅｄｃｏｌｕｍｎ処理を行う。

［１１０６］　ｉ：＝ ｉ+1 とする。

［１１０７］　ｉ ＜ Ｎ－１である場合はステップ［１１０３］に戻る。それ以外は次のステップに進む。

［１１０８］　K_{N-1}とのＸＯＲ処理を行い、RK_{N-1}をマスク値処理部に供給して１６個のＳｂｏｘ、$S_{N-1,j}(x)$（ｊ＝０，１，・・・，１５）を生成する。

［１１０９］　$S_{N-1,j}(x)$によるＳｕｂｂｙｔｅ処理、Ｓｈｉｆｔ処理、およびK_NとのＸＯＲ演算を行う。

［１１１０］　ステップ［１１０９］の演算出力に対して、マスク値生成の出力結果ＲｏｕｔとのＸＯＲ演算を行い、その結果の暗号文を出力として供給する。

【００３８】マスク値生成処理のフロー：

［１２０１］　ｉ＝０、Ｍａｓｋ＝Ｒｉｎとセットする。但し、Ｒｉｎは［１１０２］で生成された乱数マスク値である。

［１２０２］　暗号化処理部より受け取ったRK_iから、ＭａｓｋＸＯＲRK_iを演算して、新しいＭａｓｋを生成する。

［１２０３］　ステップ［１２０２］で生成した新しいＭａｓｋをＮｅｗＳｂｏｘに入力することによって、１６個のＳｂｏｘ、$S_{i,j}(x)$（ｊ＝０，１，・・・，１５）と乱数$Rout_i$を得て、$Rout_i$を新しいＭａｓｋとする。$S_{i,j}(x)$は暗号化処理部の第ｉ段のＳｕｂｂｙｔｅにおいて用いられる。

［１２０４］　ＭａｓｋをＳｈｉｆｔ処理およびＭｉｘｅｄｃｏｌｕｍｎ処理に供給し、その処理からの出力を新しいＭａｓｋとする。

［１２０５］　ｉ：＝ｉ+1とセットする。ｉ ＜ Ｎ－１である場合はステップ［１２０２］に戻る。

［１２０６］　暗号化処理部から入力されたRK_{N-1}からＭａｓｋＸＯＲRK_{N-1}を演算して、その演算結果を新しいＭａｓｋとする。

［１２０７］　Rin_{N-1}をＮｅｗＳｂｏｘに供給することによって、１６個のＳｂｏｘ、$S_{N-1,j}(x)$（ｊ＝０，１，・・・，１５）と乱数$Rout_{N-1}$を得て、$Rout_{N-1}$を新しいＭａｓｋとする。$S_{N-1,j}(x)$は暗号化処理部の第Ｎ－１段のＳｕｂｂｙｔｅにおいて用いられる。

［１２０８］　ＭａｓｋをＳｈｉｆｔ処理に供給し、その処理結果を新しいＭａｓｋとする。

［１２０９］　暗号化処理部より入力されたRK_Nから、ＭａｓｋＸＯＲRK_Nを演算し、それを出力して処

理を終了する。

【００３９】

【発明が解決しようとする課題】乱数マスク値法はＤＰＡに対する安全性（セキュリティ）が高いことが知られているが、乱数マスク値法を採用した暗号化は、その暗号化の速度が通常の暗号化と比べて数分の１と低く、非常に大きいＲＡＭ領域を必要とするという欠点がある。

【００４０】暗号化の速度が低い理由は、例えば暗号化処理におけるＸＯＲ演算では、通常の実装では２つの中間値ｘおよびｙからｚ＝ｘＸＯＲｙを演算するのに対して、乱数マスク値法ではｘ’＝ｘＸＯＲｒ$_x$、ｙ’＝ｙＸＯＲｒ$_y$を満たす中間値ｘ’、ｙ’からｚ’＝ｘ’ＸＯＲｙ’を演算する他に、ｚ’に関する新しいマスク値の生成処理ｒ$_z$＝ｒ$_x$ＸＯＲｒ$_y$を演算する必要があるからである。さらに、非線変換において、通常Ｓｂｏｘという非線形変換テーブルをＲＯＭに保持するのに対して、乱数マスク値法では、新しいマスク値から非線形変換テーブルを毎回生成しなければならないので大量の計算が要求される。

【００４１】大きいＲＡＭ領域を必要とする理由は、その非線形変換において、通常はその非線形変換テーブルをＲＯＭに保持するのに対して、乱数マスク値法では暗号化のたびに新しいＳｂｏｘをＲＡＭに格納するからである。例えば８ビット－８ビットの変換を行うＳｂｏｘを用いるラインデール法において、ＤＰＡ対策として乱数マスク値法を実装するには最低２^8＝２５６バイト分のＲＡＭ領域が必要である。しかし、例えばＳＴ１６（ＳＴマイクロエレクトロン社製）等の低コストのスマートカード用チップはＲＡＭ領域が１２８バイト程度しか持っていないので、乱数マスク値法を実装することは実際上不可能である。

【００４２】乱数マスク値法の実装のために、マスク値を共有すること、および或る暗号化から次の暗号化までの間にマスク値を生成することなどによって、見かけの処理速度の向上および必要なＲＡＭ領域の削減等が考えられているが、最初に乱数値でマスクするので全体的な処理速度の向上および大幅な必要ＲＡＭ領域の削減は不可能である。

【００４３】発明者は、暗号化において、乱数値を用いてマスクするのではなくて、固定値を用いてマスクすることによって処理速度の向上と必要なＲＡＭ領域の削減を実現すると有利であると認識した。ここで、この固定値を用いてマスクを行う方法を、固定マスク値法と呼ぶ。

【００４４】本発明の１つの目的は、共通鍵暗号を行う暗号プロセッサに対する効率的な暗号解読防止を実現することである。本発明の別の目的は、秘密鍵の推定を困難にし、暗号プロセッサの安全性を高めることである。

【００４５】

【発明の概要】本発明の１つの特徴（側面）によれば、

暗号化装置は排他的論理和手段および非線形変換手段を有する。その暗号化装置は、さらに、乱数を発生する乱数発生手段と、ｑ個（ｑは整数）の固定値と、その乱数に従ってそのｑ個の固定値の中の１つを選択する第１の選択器と、を具えている。その排他的論理和手段は、その選択された固定値と鍵の排他的論理和とその排他的論理和手段の入力の排他的論理和をとる。

【００４６】本発明の別の特徴によれば、暗号化装置は排他的論理和手段および非線形変換手段を有する。その暗号化装置は、さらに、乱数を発生する乱数発生手段と、ｑ組（ｑは整数）のマスクされた固定テーブルと、その乱数に従ってそのｑ組の固定テーブルの中の１組を選択する選択器と、を具えている。その非線形変換手段は、その選択された１組の固定テーブルに従って入力を非線形変換する。

【００４７】本発明のさらに別の特徴によれば、暗号化装置は、乱数を発生する乱数発生手段と、並列に結合された複数の暗号化部と、その乱数に従ってその複数の暗号化部の中の１つを選択する選択器と、を具えている。その複数の暗号化部の各々は、排他的論理和手段および非線形変換手段を含んでいる。

【００４８】本発明のさらに別の特徴によれば、暗号化装置は、乱数を発生する乱数発生手段と複数の暗号化段とを具えている。その複数の暗号化段の各々は、入力を非線形変換する非線形変換手段と、第１の入力と第２の入力の排他的論理和をとる排他的論理和手段と、を含んでいる。その排他的論理和手段の第２の入力はその非線形変換手段の出力に結合されている。その非線形変換手段は、ｑ個（ｑは整数）の固定値と、その乱数に従ってそのｑ個の固定値の中の１つを選択する選択器と、その選択された固定値と鍵の排他的論理和と入力の排他的論理和をとる別の排他的論理和手段と、を有する。

【００４９】本発明のさらに別の特徴によれば、暗号化装置は、乱数を発生する乱数発生手段と、並列に結合された複数の暗号化部と、その乱数に従ってその複数の暗号化部の中の１つを選択する選択器と、を具えている。その複数の暗号化部の各々は複数の暗号化段を有する。その複数の暗号化段の各々は、入力を非線形変換する非線形変換手段と、第１の入力と第２の入力の排他的論理和をとる排他的論理和手段と、を含んでいる。その排他的論理和手段の第２の入力はその非線形変換手段の出力に結合されている。

【００５０】本発明のさらに別の特徴によれば、暗号化装置において用いられるプログラムは、乱数に従って複数の暗号化手順の中の１つを選択するステップと、その選択された暗号化手順に従って、入力値を暗号化して出力するステップと、を含んでいる。その暗号化するステップは、固定値と鍵の排他的論理和と入力値の排他的論理和をとるステップと、１組の固定テーブルに従って入力値を非線形変換するステップと、を含んでいる。

【００５１】本発明によれば、共通鍵暗号を行う暗号プロセッサに対する効率的なＤＰＡ対策を実現でき、ＤＰＡを用いた秘密鍵の解読（analyze）を困難にし、暗号プロセッサの安全性を高めることができる。

【００５２】実施形態においては、通常は乱数値で入力や鍵をマスクする乱数マスク値法に対して、本発明の固定マスク値法に従って、複数個の固定値を用意し、それを乱数によって切換えることによって、乱数マスク値法と同様の効果を得る。固定マスク値法では、マスク値が特定の固定値に限定されるので、新しいマスク値を予め求めておくことによって処理速度を高くできる。また、予め固定マスク値の集合（セット）を用意し、それぞれの固定マスク値に対応した非線形変換テーブルをＲＯＭに用意することによって、小さいＲＡＭ領域を有するプラットフォーム上でも実装が可能になる。例えばＳＴ１６等の低コストのスマートカード用のＬＳＩチップでも６Ｋバイト程度の大きなＲＯＭ領域を備えているので、固定値マスク法は低コストのスマートカードに適している。

【００５３】

【発明の好ましい実施形態】次に、本発明の実施形態を説明する。実施形態において、暗号化における全てまたは一部の諸要素および諸処理は、集積回路等によってハードウェアの形態で実装してもよいし、またはプロセッサによって実行されるプログラムの形態で実装してもよい。

【００５４】図２１は本発明による第１のタイプの暗号化装置１００の概略的構成を示している。図２２および２３は、暗号化装置１００の鍵ＸＯＲ演算および非線形変換の構成をそれぞれ示している。線形変換は図２に示されているものを用いればよい。

【００５５】図２１において、暗号化装置１００は、乱数Ｒ（Ｒ＝０，１，・・・，ｑ－１）を発生する乱数発生器１０３と、その乱数に従って固定マスク値$FM_{0,R}$の中の１つを選択して供給する切換部または選択器１０４と、入力の平文と選択された固定マスク値$FM_{0,R}$との排他的論理和を演算する（ＸＯＲする）（Ｘｉｎ’＝平文ＸＯＲ$FM_{0,R}$）排他的論理和１０６と、入力Ｘｉｎ’を受取り乱数Ｒに従って入力Ｘｉｎ’を暗号化して出力Ｘｏｕｔ’を生成する暗号化処理部１０１と、その乱数に従って固定マスク値$FM_{N+1,R}$の中の１つを選択して供給する切換部１０５と、その出力Ｘｏｕｔ’と選択された固定マスク値$FM_{N+1,R}$との排他的論理和を演算して（ＸＯＲして）暗号文（Ｘｏｕｔ’ＸＯＲ$FM_{N+1,R}$）を生成する排他的論理和１０７と、を含んでいる。ここで、Ｎは暗号化処理部１０１で使用される固定マスク値のセットの数を表す。

【００５６】暗号化装置１００は、さらに、作業メモリ用のＲＡＭ１６２と、固定マスク値と、固定非線形変換テーブルＳｂｏｘと、線形変換関数Ｌ等を格納するＲＯＭ１６４と、ＲＯＭ等のプログラム・メモリ１６０に格納されているプログラムに従ってこれらの諸処理要素１０３～１０７を制御するプロセッサ１５０とを含んでいる。代替構成として、プロセッサ１５０は、諸処理要素１０３～１０７に対応する機能を実装したメモリ１６０中のプログラムを実行することによって諸処理要素１０３～１０７を実現してもよい。この場合、図２１はフロー図として見ることができる。

【００５７】暗号化処理部１０１において、図２２の鍵ＸＯＲ、図２の線形変換および図２３の非線形変換の組み合わせによって規定されるラウンド関数が繰返し実行され、または縦続接続されたそのような複数のラウンド関数回路によってそのラウンド関数が実行される。

【００５８】図２２の鍵ＸＯＲは、切換部１０９によって乱数Ｒに従って選択された固定マスク値$FM_{i,R}$と鍵K_iとをＸＯＲして或る値を供給し、その値と入力Ｘｉ’とをＸＯＲして出力Ｚｉ’を供給する。図２３の非線形変換は、切換部１１１～１１９によって乱数Ｒに従ってＳｂｏｘの要素$wi_{j,R}$’（ｊ＝０，・・・，ｕ－１）を選択し、その選択された要素$wi_{j,R}$’を用いて非線形変換する。

【００５９】乱数マスク値法において示した式（７）の関係を用いると、図２２の鍵ＸＯＲにおいて図１１のＲｉとＲＯｉに対応して次の式（８）が得られる。

【数８】

$$ROi = Ri \oplus FM_{i,R} \qquad (8)$$

【００６０】図２３の非線形変換において用いられる図１３（Ｂ）のＮｅｗＳｂｏｘのｒｉとｒｏｉに対応して次の式（９）が得られる。

【数９】

$$wi'_{i,R}(xi') = wi(ri \oplus xi') \oplus roi \qquad (9)$$

【００６１】図２４は本発明による第２のタイプの暗号化装置２００の概略的構成を示している。図２５および２６はその鍵ＸＯＲおよび非線形変換の構成をそれぞれ示している。線形変換はいずれの暗号化処理部においても図２に示されているものを用いればよい。

【００６２】図２４において、暗号化装置２００は、乱数Ｒを発生する乱数発生器２０３と、その乱数に従って固定マスク値$FM_{0,R}$の中の１つを選択して供給する切換部２０４と、入力の平文と切換部２０４によって選択された固定マスク値$FM_{0,R}$とをＸＯＲする（Ｘｉｎ’＝平文ＸＯＲ$FM_{0,R}$）ＸＯＲ（排他的論理和）２０６と、入力Ｘｉｎ’を受取りその入力Ｘｉｎ’を暗号化して出力Ｘｏｕｔ’を生成する複数の暗号化処理部２０８～２０９と、入力Ｘｉｎ’を受取り乱数Ｒに従って互いに並列に結合された暗号化処理部２０８～２０９の中の１つを選択してその選択された暗号化処理部に入力Ｘｉｎ’を供給する切換部２１１と、同じ乱数Ｒに従って暗号化処理部２０８～２０９の中の同じ１つを選択してその選択された暗号化処理部からの出力Ｘｏｕｔ’

を供給する切換器２１３と、その乱数に従って固定マスク値$FM_{N+1,R}$の中の１つを選択して供給する切換部２０５と、その出力Ｘｏｕｔ’と選択された固定マスク値$FM_{N+1,R}$とをＸＯＲして暗号文（＝Ｘｏｕｔ’ＸＯＲ$FM_{N+1,R}$）を生成するＸＯＲ２０７と、を含んでいる。ここで、Ｎは使用される固定マスク値の数を表す。図２４において、２つの切換部２１１および２１３のうちの一方だけで選択してもよく、その他方の切換部を省いてもよい。

【００６３】暗号化装置２００は、さらに、作業メモリ用のＲＡＭ２６２と、固定マスク値、固定非線形変換テーブルＳｂｏｘと、線形変換関数Ｌ等を格納するＲＯＭ２６４と、ＲＯＭ等のプログラム・メモリ２６０に格納されているプログラムに従ってこれらの諸処理要素２０３～２１１を制御するプロセッサ２５０とを含んでいる。代替構成として、プロセッサ２５０は、諸処理要素２０３～２１１に対応する機能を実装したメモリ２６０中のプログラムを実行することによって諸処理要素２０３～２１１を実現してもよい。この場合、図２４はフロー図として見ることができる。

【００６４】暗号化処理部２０８～２０９は、固定マスク値$FM_{i,j}$およびＳｂｏｘのセット$wi_{i,j}$が異なること以外は構成が共に同じである。各暗号化処理部２０８～２０９において、図２５の鍵ＸＯＲ演算、図２の線形変換処理および図２６の非線形変換の組み合わせによって規定されるラウンド関数が繰返し実行され、または縦続に結合されたそのような複数のラウンド関数回路によってそのラウンド関数が実行される。

【００６５】図２５の鍵ＸＯＲは、暗号化処理部２０８～２０９の各々に固有の固定マスク値$FM_{i,R}$と鍵ＫｉをＸＯＲして或る値を供給し、その値と入力Ｘｉ’とをＸＯＲして出力Ｚｉ’を供給する。図２６の非線形変換は、その暗号化処理部に固有のＳｂｏｘのセット$wi_{j,R}$’（ｊ＝０，・・・，ｕ－１）を用いて非線形変換する。ここで、乱数マスク値法における式（７）の関係を用いると、図２５の鍵ＸＯＲにおいて図１１のＲｉとＲＯｉに対応してＲＯｉ＝ＲｉＸＯＲ$FM_{i,R}$が得られ、図２６の非線形変換において図１２（Ｂ）のｒｉとｒｏｉに対応して$wi_{j,R}$’（ｘｉ’）＝ｗｉ（ｒｉＸＯＲｘｉ’）ＸＯＲｒｏｉが得られる。

【００６６】本発明による上述の固定マスク値法は、低コストのスマートカードに実装するのに適した方法と考えられる。ここでは、次のような問題点を検討することによってより好ましい実装形態を実現する。

【００６７】（Ａ）ＤＰＡ対策としての有効性の問題

乱数マスク値法はＤＰＡに対して安全であることが知られているが、固定マスク値法はＤＰＡに対してどの程度安全であるかが不明であり、実装法および固定マスク値の条件しだいではＤＰＡに対して弱い可能性がある。

【００６８】（Ｂ）ＲＯＭ領域が膨大になる可能性の問題

固定マスク値法は、非線形変換テーブルをＲＯＭに用意することで、乱数マスク値法より必要なＲＡＭ領域を小さく抑えることができるという利点がある。しかし、用意されるマスク固定値の数に応じて使用ＲＯＭ量が決まる。安全性を確保するために必要なＲＯＭの容量が膨大になる可能性がある。

【００６９】（Ａ）の問題に関連して、本発明を適用する或る暗号化では、或る条件の固定マスク値を使用すればその数ｑが２でも充分に安全であることが分かっている。本発明による暗号化法としてラインデール法を適用した場合の安全性を後で説明する。同様に（Ｂ）の問題に関連して、相異なる固定マスク値の数ｑを２程度にして、同じラウンドやラウンド毎のＳｂｏｘを共通にすることで、Ｓｂｏｘのデータ量を、本発明が適用されない通常の実装の約２～数倍に抑えて、使用ＲＯＭ量を小さく抑える。実施形態において暗号化法としてラインデール法を適用した場合の使用ＲＯＭ量について後で述べる。

【００７０】図２７は、図２１の第１のタイプの暗号化装置１００の一例を示しており、ラインデール（Rijndael）暗号化法に固定マスク値法を適用した暗号化装置３００を示している。但し、図２７において、図２１におけるプロセッサ１５０、メモリ１６０、１６２および１６４は簡単化のために省略されている。図２７において、暗号化装置３００は、乱数ｈ（ｈ＝０，１，・・・，ｑ－１）を発生する乱数発生器３０３と、その乱数ｈに従って固定マスク値$FMinn_h$の中の１つを選択して供給する選択器３０５と、入力の平文と選択された固定マスク値$FMinn_h$とをＸＯＲするＸＯＲ３０２と、入力Ｘｉｎ’を受取って乱数ｈに従って入力Ｘｉｎ’を暗号化して出力Ｘｏｕｔ’を生成する複数段の暗号化処理段３１０（各段ｉは０≦ｉ≦Ｎ－２）と、第Ｎ－２段の暗号化処理段３１０の出力Ｘｏｕｔ’を入力Ｘｉｎ’として受取り乱数Ｒに従ってその入力を暗号化して暗号文Ｘｏｕｔ’を生成する第Ｎ－１段の暗号化処理段３１１と、を含んでいる。

【００７１】複数段の暗号化処理段３１０の各段ｉは、乱数ｈに従って選択された固定マスク値$FM_{i,h}$を供給する選択器３２９と、鍵K_iと固定マスク値$FM_{i,h}$をＸＯＲして出力を供給するＸＯＲ３３１、その出力の値と入力Ｘｉ’とをＸＯＲするＸＯＲ３３３と、その乱数ｈに従ってＳｂｏｘである固定値$S_{i,j,h}$（ｊ＝０，１，・・・，１５）の中の１つを選択してマスクされた鍵$K_{0,h}$を供給する切換器または選択器３３９と、その選択された$S_{i,j,h}$に従ってＸＯＲ３３３からの出力をサブバイトするＳｕｂｂｙｔｅ３３４と、Ｓｕｂｂｙｔｅ３３４の出力をシフトするＳｈｉｆｔ３３５と、Ｓｈｉｆｔ３３５の出力をミクストコラムするＭｉｘｅｄＣｏｌｕｍｎ３３６とを含んでいる。

17

【００７２】第Ｎ－１段の暗号化処理段３１１は、第ｉ段の暗号化処理段３１０と同様の切換器３２９および３３９と、ＸＯＲ３３１および３３３と、Ｓｕｂｂｙｔｅ３３４と、Ｓｈｉｆｔ３３５とを有し、ＭｉｘｅｄＣｏｌｕｍｎは有せず、切換器３７９によって選択された固定マスク値$FM_{N,h}$とK_NとをＸＯＲ３７１によってＸＯＲしてマスクされた鍵$K_{N,h}$を生成し、$K_{N,h}$とＳｈｉｆｔ３３５の出力とをＸＯＲ３７３によってＸＯＲし、切換器３９９によって選択された固定マスク値$FMout_h$とＸＯＲ３７３の出力とをＸＯＲ３８３によってＸＯＲして暗号文を生成する。

【００７３】ここで、$FM_{i,h}$、$FMin_h$、および$FMout_h$は、固定マスク値である。$S_{i,j,h}$は、固定のＳｂｏｘである。これらの固定マスク値およびＳｂｏｘ値は予め求められる。従って、固定マスク値法では通常の乱数マスク値法における逐次計算が不要なので、処理が高速化される。さらに、予め求めた固定マスク値およびＳｂｏｘの変換テーブルを、例えば図２１のＲＡＭ１６２および図２４のＲＡＭ２６２のようなＲＡＭにではなくて、図２１のＲＯＭ１６４および図２４のＲＯＭ２６４のようなＲＯＭに格納することによって、実装に必要なＲＡＭ領域を大幅に節約できる。このような必要なＲＡＭ量の節約は、ほんの１２８バイトのＲＡＭ領域しか持っていない低コストのスマートカードへの実装に有利である。

【００７４】図２７の暗号化装置３００に入力平文が供給されると、内部の乱数発生器３０３によって$0 \leqq h \leqq q-1$を満たす乱数ｈが生成される。その生成された乱数ｈに従って、図２７に示す全ての切換器３０５、３２９、３３９、３７９および３９９において、ｑ個の入力値の中からｈ番目のものが選択されて供給される。図２８は図２７におけるＳｕｂｂｙｔｅ３３４の構成を例示している。Ｓｈｉｆｔ３３５は図１６に例示されているものを、Ｍｉｘｅｄｃｏｌｕｍｎ３３６は図１７に例示されているものを用いる。図２８のＳｕｂｂｙｔｅにおいて用いられるＳｂｏｘ　$S_{i,j,h}$は、図１６に示されている通常のラインデール法のＳｕｂｂｙｔｅのＳｂｏｘであるＳを用いて次の式（１０）で表される。

【数１０】

$$S_{i,j,h}(x) = S(x \oplus a_{i,j,h}) \oplus b_{i,j,h} \qquad (10)$$

【００７５】式（１０）における$a_{i,j,h}$および$b_{i,j,h}$は、式（９）の$ri_{i,j}$および$roi_{i,j}$にそれぞれ対応し、それぞれｉ番目のラウンドのＳｕｂｂｙｔｅの入力および出力のマスク値に対応する。$a_{i,j,h}$は、入力マスク値なので、そのＳｕｂｂｙｔｅより前のマスク値および処理等によって一意的に決まる。一方、$b_{i,j,h}$は任意に設定可能である。

【００７６】図２７の暗号化装置３００処理のフローをステップ［１３０１］～［１３１４］に示す。ステップ

18

［１３０３］～［１３０９］は図２８の第ｉ段における処理である。ステップ［１３１０］～［１３１４］は図２７の第Ｎ－１段における処理である。

［１３０１］　ｉ＝０とセットする。

［１３０２］　入力平文を受け取り、次いで乱数発生器３０３によって乱数ｈ（$0 \leqq h \leqq q-1$）が発生される。この乱数ｈは後のステップで使用される。

［１３０３］　入力平文に対して、固定マスク値の集合$\{FMin_0, \cdots, FMin_{q-1}\}$の中から$FMin_h$を切換器３０５によって選択した後、ＸＯＲ３０２によって入力平文と$FMin_h$をＸＯＲする。ＸＯＲ３０２の出力を中間データＸとする。

［１３０４］　固定マスク値の集合$\{FM_{i,0}, \cdots, FM_{i,q-1}\}$の中から$FM_{i,h}$を切換器３２９によって選択して、拡大鍵$K_i$、中間データＸおよび$FM_{i,h}$に対してＸＸＯＲ$K_i$ＸＯＲ$FM_{i,h}$を演算し、その演算結果を新しい中間データＸとする。

［１３０５］　Ｓｕｂｂｙｔｅ３３４において、中間データＸに対して、切換器３３９によって乱数ｈに従って選択された非線形変換テーブル$S_{i,j,h}$（ｘ）に従ってサブバイト処理する。その演算結果を新しい中間データＸとする。

［１３０６］　Ｓｈｉｆｔ３３５において中間データＸをシフトする。そのシフトされたデータを新しい中間データＸとする。

［１３０７］　Ｍｉｘｅｄｃｏｌｕｍｎ３３６において中間データＸをミクストコラム処理する。その処理結果を新しい中間データＸとする。

［１３０８］　ｉ：＝ｉ＋１とセットする。

［１３０９］　ｉ ＜ Ｎ－１の場合は［１３０３］に戻る。それ以外は次のステップに進む。

［１３１０］　乱数ｈに従って固定マスク値の集合$\{FM_{N-1,0}, \cdots, FM_{N-1,q-1}\}$の中から$FM_{N-1,h}$を切換器３２９によって選択して、拡大鍵$K_{N-1}$、中間データＸおよび$FM_{N-1,h}$に対して、ＸＸＯＲ$K_{N-1}$ＸＯＲ$FM_{N-1,h}$を演算する。その演算結果を新しい中間データＸとする。

［１３１１］　切換器３３９によって選択された非線形変換テーブル$S_{N-1,j,h}$（ｘ）に従ってサブバイト処理を行う。その処理結果を新しい中間データＸとする。

［１３１２］　Ｓｈｉｆｔ３３５において中間データＸをシフトする。そのシフトされたデータを新しい中間データＸとする。

［１３１３］　乱数ｈに従って固定マスク値の集合$\{FM_{N,0}, \cdots, FM_{N,q-1}\}$の中から切換器３７９によって$FM_{N,h}$を選択して、その中間データＸ、拡大鍵$K_N$および$FM_{N,h}$に対して、ＸＸＯＲ$K_N$ＸＯＲ$FM_{N,h}$を演算する。その演算結果を新しい中間データＸとする。

19

［１３１４］　乱数ｈに従って固定マスク値の集合｛ＦＭｏｕｔ０，・・・，ＦＭｏｕｔｑ－１｝の中からＦＭｏｕｔｈを切換器３９９によって選択して、中間データＸとＦＭｏｕｔｈをＸＯＲし、その演算結果を出力暗号文Ｘｏｕｔ’として供給する。

【００７７】図２９は、ラインデール法に固定マスク値法を適用した別の暗号化装置４００を例示している。但し、図２９において、図２１におけるプロセッサ１５０、メモリ１６０、１６２および１６４は簡単化のため*

20

*に省略されている。図２９の構成は、その各ラウンド関数における各Ｓｕｂｂｙｔｅ３３４に結合された切換器３３９に供給されるＳｂｏｘが互いに同じである以外は、図２７と同じであり、同じ構成要素については再び説明はしない。図３０は図２９におけるＳｕｂｂｙｔｅ３３４の構成を例示している。マスク値ＦＭｉ，ｈは、任意のｈ＝０，１，・・・，ｑ－１について次の式（１１）を満たす。

【数１１】

$$FM_{ih}=\begin{cases} C_h \oplus FMin_h & (i=0) \\ C_h \oplus D_h & (i=1,\ldots,N-1) \\ D_h & (i=N) \end{cases} \qquad (11)$$

【００７８】ここで、ＣｈおよびＤｈは、１６バイトの定数であり、Ｉバイト定数ｃｈ，ｊおよびｄｈ，ｊ（ｊ＝０，１，・・・，１５）をそれぞれ用いて、次の式※（１２）のように記述される。

【数１２】

$$\begin{cases} C_h = c_{h,15} \ldots c_{h,1} c_{h,0} \\ D_h = d_{h,15} \ldots d_{h,1} d_{h,0} \end{cases} \qquad (12)$$

【００７９】図２９の暗号化装置４００において、サブバイト処理Ｓｕｂｂｙｔｅ３３４は、切換器３３９によって選択された１６個のＳｂｏｘ、Ｓ０，ｈ、Ｓ１，ｈ、・・・、Ｓ１５，ｈを用いて、図３０に示す非線形変換処理を行う。１６個のＳｊ，ｈはＤＰＡ対策なしの通常のラインデール法におけるＳであり、および式（１２）におけるｃｈ，ｊ、ｄｈ，ｊを用いて、Ｓｊ，ｈ（ｘ）＝Ｓ（ｘＸＯＲｃｈ，ｊ）ＸＯＲｄｈ，ｊと表される。

【００８０】図２９の暗号化装置４００の処理フローにおいては、図２７の暗号化装置４００の処理フローにおけるステップ［１３０５］および［１３１１］において、切換器３３９によって乱数ｈに従って１６個のＳｂｏｘ、Ｓ０，ｈ、Ｓ１，ｈ、・・・、Ｓ１５，ｈの中から選択された非線形変換テーブルＳｊ，ｈに従ってサブバイト処理する。

【００８１】図２７の暗号化装置３００ではＳｕｂｂｙｔｅ３３４のテーブルが各ラウンドで異なるのに対して、図２９の暗号化装置４００では全てのラウンドで共通のテーブルが使用される。これが可能な理由を説明する。まず、通常のラインデール処理におけるＳｕｂｂｙ★te（図１６）への入力値をＸと表す。これに対し、図３０のＳｕｂｂｙｔｅ処理の入力値はＸＸＯＲＣｈと表すことができる。これは、マスク値の間に式（１１）の関係が成立するからである。Ｃｈは段数ｉとは関係ない定数なので、式（１０）におけるＳｉ，ｊ，ｈ（ｘ）＝Ｓ（ｘＸＯＲａｉ，ｊ，ｈ）ＸＯＲｂｉ，ｊ，ｈに対して、ａｉ，ｊ，ｈを段番号ｉに関係ない定数に設定できる。また、ｂｉ，ｊ，ｈは、任意の定数なので、ｉに関係ない定数に設定できる。よって、図３０に示すようにＳｊ，ｈ（ｘ）＝Ｓ（ｘＸＯＲｃｈ，ｊ）ＸＯＲｄｈ，ｊで表される段数ｉに無関係なＳｂｏｘを用いたサブバイト処理が可能となる。従って、図２９の暗号化装置４００で用いられるＳｂｏｘに必要な使用ＲＯＭ量は、図２７の暗号化装置３００の場合の１／Ｎとすることができる。

$$\begin{cases} c_{h,15} = c_{h,14} = \ldots = c_{h,0} \\ d_{h,15} = d_{h,14} = \ldots = d_{h,0} \end{cases} \qquad (13)$$

【００８２】さらに、図２９の暗号化装置４００において、マスク値ＣｈおよびＤｈに関する条件式（１２）に、次の式（１３）に示す条件を加えることによって、使用するＳｂｏｘの数（種類の数）を相異なる１６セットからほんの１セットに減らすことができる。

【数１３】

【００８３】即ち、図２９の暗号化装置４００で用いられるＳｂｏｘに必要な使用ＲＯＭ量は、図２７の暗号化装置３００の場合の１／１６とすることができる。以上から、式（１３）を満たす図２９の暗号化装置４００を用いることによって、Ｓｂｏｘに必要なＲＯＭ領域を図２７の暗号化装置３００を用いた場合の1／（１６Ｎ）とすることができる。

【００８４】また、図２９の暗号化装置４００では入力側および出力側においてＦＭｉｎｈおよびＦＭｏｕｔｈをそれぞれＸＯＲしているが、この処理は安全性に寄与しないことが判明しているので、省略できる。さらに、マスク済みの鍵ＫｉＸＯＲＦＭｉ，ｈの値を用いることによって、鍵Ｋｉと固定マスク値ＦＭｉ，ｈとのＸＯＲ演算を省略できる。これらの省略によって、切換器によ

る処理が少し増加するだけなので、ＤＰＡ対策のないラインデール法の場合とほぼ同じ計算量で固定マスク値法を適用したラインデール法が実現できる。

【００８５】図３１は、図２４の第２のタイプの暗号化装置２００の一例を示しており、ラインデール法に固定マスク値法を適用した別の暗号化装置５００を例示している。但し、図３１において、図２４におけるプロセッサ２５０、メモリ２６０、２６２および２６４は簡単化のために省略されている。図３１において、暗号化装置５００は、乱数ｈを発生する乱数発生器５０３と、乱数ｈに従って切換えを行う切換器５０２および５０４と、切換器５０２および５０４によって乱数ｈに従って選択される互いに並列に結合された第０～第ｑ－１のｑ個の暗号処理部５１１～５１３と、を含んでいる。

【００８６】暗号処理部５１１～５１３の各々は、入力Ｘｉｎ’を受け取って出力Ｘｏｕｔ’を生成する複数段の暗号化処理段５３０（段ｉは０≦ｉ≦Ｎ－２）と、前段の出力を入力として受取りその入力を暗号化して出力Ｘｏｕｔ’を生成する第Ｎ－１段の暗号化処理段と５３１、を含んでいる。第０段～第Ｎ－２段の暗号化処理段５３０は、固定マスク値、ＸＯＲ５２３、それぞれのＳｕｂｂｙｔｅ５２５、Ｓｈｉｆｔ５２６およびＭｉｘｅｄｃｏｌｕｍｎ５２７を有し、それぞれの固定マスク値および固定Ｓｂｏｘに従って暗号化を行う。第０段の暗号化処理段５３０の前にはＸＯＲ５２１が結合されている。第Ｎ－１段の暗号化処理段５３１は、固定マスク値、ＸＯＲ５２３、５２８および５２９、Ｓｕｂｂｙｔｅ５２５およびＳｈｉｆｔ５２６を有し、それぞれの固定マスク値および固定Ｓｂｏｘに従って暗号化を行う。図３１では、鍵K_iと$FM_{j,h}$とがＸＯＲされたK_i ＸＯＲ$FM_{j,h}$の値が直接供給されるように示されているが、図２９の場合と同様に鍵K_iと$FM_{j,h}$とがＸＯＲによってＸＯＲされてＸＯＲ５２３～５５３の入力に供給されるようにしてもよい。図３１においては、図２９の場合と同様に$FMin_h$と$FMout_h$とを省略できる。

【００８７】図３１の暗号化装置５００においても、図２９の暗号化装置４００の場合と同様に全てのラウンドで同じＳｂｏｘを使用しており、式（１３）の条件を満*

*たすようにすれば、図２９の暗号化装置４００の場合と同様のＲＯＭの削減が可能となる。暗号化装置５００の計算量は、暗号化装置４００の場合より切換器が少ないだけ有利であり、ＤＰＡ対策のないラインデール法とほぼ同じ計算量となる。但し、図３１の暗号化装置５００は、図２９の暗号化装置４００の場合よりもシフトおよびミクストコラム処理の数が増えるので、暗号化装置５００の方が回路のサイズが大きくなる。なお、暗号化装置５００には２つの選択器５０２および５０４が存在するが、左右の切換器５０２および５０４のうち一方だけで切換えてもよい。そのような場合、他方の選択器は省略できる。

【００８８】図２９および図３１の暗号処理装置においては同じ処理が異なる構成で行われるだけなので、安全性は共に同じである。

【００８９】次に、固定マスク値法の安全性について説明する。図２７の暗号化装置では、固定マスクの数ｑが充分大きい場合は、乱数マスク値法と実質的に同じ動作になるので、同様に高い安全性を有する。乱数マスク値法におけるほんの１ラウンドに簡略化された暗号関数について安全性が証明できるので、各ラウンドに同じＳｂｏｘを使用する図２９および３１の暗号化装置４００および５００についても同様に安全性が証明できる。

【００９０】次に、ｑが小さい場合の固定マスク値法の安全性を説明する。ｑ＝１では安全でないことが分かっている。次に小さいｑ＝２の場合の安全性を評価する。図２９および３１の暗号化装置４００および５００において、ｑ＝２とし、式（１３）に従ってＳｕｂｂｙｔｅにおいて１セット（種類）だけのＳｂｏｘを用い、かつ$FMin_h$および$FMout_h$を省略した最も単純な場合においても、$c_{0,j}$、$c_{1,j}$、・・・、$c_{q-2,j}$に関して次の式（１４）または式（１５）の条件を設定し、$d_{0,j}$、$d_{1,j}$、・・・、$d_{q-2,j}$に関して次の式（１６）の条件に設定することによって、ＤＰＡに対する安全性を高めることができる。

【００９１】ｑ＝２、Ｓｂｏｘのセット（種類）の数が１、全てのｊ＝０、１、・・・、１５について

【数１４】

$$c_{0,j} \oplus c_{1,j} = (10101010)_2 \text{ または } (01010101)_2 \qquad (14)$$

ｑ≧２、Ｓｂｏｘのセットの数が１、全てのｊ＝０、１、・・・、１５について ※

※【数１５】

$$(c_{0j} \oplus c_{1j}) \vee (c_{1j} \oplus c_{2j}) \vee \ldots \vee (c_{q-2j} \oplus c_{q-1j}) = (11111111)_2 \qquad (15)$$

ｑ≧２、Ｓｂｏｘ数が１、全てのｊ＝０、１、・・・、１５について ★

★【数１６】

$$(d_{0j} \oplus d_{1j}) \vee (d_{1j} \oplus d_{2j}) \vee \ldots \vee (d_{q-2j} \oplus d_{q-1j}) = (11111111)_2 \qquad (16)$$

ここで、（ ）$_2$は２進法の値を表す。

【００９２】ＤＰＡは、図９の測定対象点Ａにおける所定タイミングで行われる場合と、測定対象点ＢおよびＣにおける所定タイミングで行われる場合とがある。図２

９および３１の暗号化装置４００および５００が安全であることを説明する。ビット（ｘ，ｅ）はｘの第ｅ桁のビット値を表す。

【００９３】攻撃者は、鍵の推定のために以下の（ｉ）および（ｉｉ）の処理を行う。

（ｉ）ＤＰＡを用いて、チェックする仮の鍵（可能な鍵）の数を限定する。

（ｉｉ）（ｉ）で限定された数の仮の鍵について、プロセッサ内部で用いられている真の鍵と値が一致するかどうかをチェックすることによって鍵を推定する。１つの仮の鍵すなわち１つのパターンの鍵をチェックするための計算の量を１単位（サイクル）とする。

【００９４】上述の鍵の値のチェックは、暗号プロセッサの平文と暗号文の関係を調べることによって実現できる。即ち、暗号化プロセッサにより平文を暗号化した場合と、ソフトウェアなどの別の手段によって、そのチェックされる仮の鍵の値を用いてその平文を暗号化した場合とを比較する。平文と暗号文の関係が両方の暗号化で一致した場合は、その仮の鍵の値が暗号プロセッサ内部で用いられていると判定する。両者が一致しない場合はその仮の鍵は用いられていないと判定する。

【００９５】ＤＰＡを用いて１２８ビットの鍵を推定する

*るのに必要な計算量の例を示す。例えば、ＤＰＡを行っても１２８ビット鍵に関する有益な情報を得られなかった場合は、１２８ビット鍵の全ての可能なパターンをチェックする必要があるので、必要な計算量は2^{128}単位である。例えば、ＤＰＡによって１２８ビット鍵の最下位ビット（ＬＳＢ）が０であることが分かった場合は、その鍵の残りの１２７上位ビット（ＭＳＢｓ）の全ての可能なパターンをチェックすればよいので、必要な計算量は2^{127}単位である。

【００９６】次に、Ｓｂｏｘの出力のＤＰＡに対する安全性について、図９の測定対象点Ａにおけるを図２９および３１の暗号化装置４００および５００に対して適用した場合について説明する。図２９および３１の第０段のＳｕｂｂｙｔｅにおいて、それぞれのＳｂｏｘの出力値がロードされるタイミングにおいてＤＰＡを適用すると、鍵K_iは$2^{8(16-F)}=2^{128-8F}$に比例した計算量で解読できることが分かっている。ここで、$F=f_0+f_1+\cdots+f_{15}$である。f_jは、次のように定義される。ｊ（ｊ＝０，１，・・・，１５）番目のＳｂｏｘの出力マスク値を$d_{0,j}$、$d_{1,j}$、・・・、$d_{q-1,j}$としたときに、

【数１７】

$$WD_j=(d_{0,j}\oplus d_{1,j})\vee(d_{1,j}\oplus d_{2,j})\vee\cdots\vee(d_{q-2,j}\oplus d_{q-1,j})$$

について、

$$\begin{cases} WD_j=(11111111)_2\text{ならば、}f_j=0、\\ \text{それ以外の場合は、}f_j=1、\end{cases}$$

【００９７】よって、ｊ＝０，１，・・・，１５の全てについて$f_j=0$の場合に、暗号化装置はＤＰＡに対して最も安全である。そのためには、全てのｊ＝０，１，・・・，１５について、$WD_j=(11111111)_2$でなければならない。このとき、ＤＰＡを用いて鍵を求めるのに必要な計算量は最大の2^{128}となる。

【００９８】次に、鍵ＸＯＲの出力（図９の測定対象点Ｂ）またはＳｂｏｘへの入力（図９の測定対象点Ｃ）におけるＤＰＡに対する安全性を説明する。図９の測定対象点ＢおよびＣにおける所定タイミングにおけるＤＰＡに対する安全性は、暗号プロセッサでＲＡＭへのロードが行われた時の測定電圧とロード値との関係がどのようなモデルを用いて近似できるかによって異なる。まず、任意モデルの場合のＤＰＡを考え、その後で式（６）で表される隣接ビット・モデルの場合のＤＰＡを考える。

【００９９】任意モデルの場合について、図９の測定対象点BおよびCにおける所定タイミングにおけるＤＰＡを、式（１３）によってサブバイト処理で用いられるＳｂｏｘの数を１つ（１セット）だけとした図２９および３１の暗号化装置４００および５００に対して適用した場合について説明する。第０段のＳｕｂｂｙｔｅ処理において、それぞれの鍵ＸＯＲの出力値がロードされるタイミングにおいてＤＰＡを適用すれば、鍵K_iは$2^{128-(15/16)H}$に比例した計算量で解読できることが分かっている。ここで、$H=h_0+h_1+\cdots+h_{15}$である。h_jは次のように定義される。ｊ（ｊ＝０，１，・・・，１５）番目のＳｂｏｘの入力マスク値を$c_{0,j}$、$c_{1,j}$、・・・、$c_{q-1,j}$としたときに、

【数１８】

$$WC_j = (c_{0,j} \oplus c_{1,j}) \vee (c_{1,j} \oplus c_{2,j}) \vee \cdots \vee (c_{q-2,j} \oplus c_{q-1,j})$$ について、

$$WC_j = (wc_{j,7} wc_{j,6} wc_{j,5} wc_{j,4} wc_{j,3} wc_{j,2} wc_{j,1} wc_{j,0})_2$$ において、

$$h_j = (e = 0, 1, \cdots, 7 \text{に関して} wc_{j,e} = 0 \text{が成立するeの個数})$$

【０１００】即ち、ｊ＝０，１，・・・，１５についてh_j＝０の場合がＤＰＡに対して最も安全である。そのためには、全てのｊ＝０，１，・・・，１５についてWC_j＝（１１１１１１１１）$_2$でなければならない。このとき、ＤＰＡを用いて鍵を求めるのに必要な計算量は最大の2^{128}になる。

【０１０１】次に、隣接ビット・モデルの場合について、ｑ＝２かつ式（１３）によりサブバイト処理で用いるＳｂｏｘのセット（種類）の数を１に限定し、式（６）で表される隣接ビット・モデルを適用した場合について説明する。隣接ビット・モデルは、低コストのスマートカードにおける電圧を近似するのに適したモデル*として知られており、このモデルが適用できる場合は、上述の任意モデルでは解読できない鍵情報を解読できる。図２９および３１の暗号化装置４００および５００における第０段のＳｕｂｂｙｔｅ処理において、それぞれのＳｂｏｘの入力値がロードされるタイミング（図９の測定対象点Ｃにおける所定タイミング）にＤＰＡを適用することによって、鍵K_iを$2^{128-(15/16)H}$に比例した計算量で解読できる。ここで、$H = h_0 + h_1 + \cdots + h_{15}$である。$h_j$は、次のように定義される。ｊ番目のＳｂｏｘの入力マスク値を$c_{0,j}$および$c_{1,j}$としたときに、

【数１９】

$$WC_j = c_{0,j} \oplus c_{1,j} = (wc_{j,7} wc_{j,6} \cdots wc_{j,0})_2 \text{ において、}$$

$$h_j = (e = 0, 1, \ldots, 7 \text{に関して} wc_{j,e} = 0 \text{を満たすeの個数}) +$$

$$(e = 0, 1, \ldots, 7 \text{に関して} wc_{j,e} = wc_{j,e+1} = 1 \text{を満たすeの個数})$$

ここで、$wc_{j,e}$＝０または１である。ｊはＳｂｏｘの序数（番号）を表す。ｅはビット位置を表す。

【０１０２】Ｓｂｏｘの数は１なので、条件式（１４）が$c_{0,j}$ＸＯＲ$c_{1,j}$＝（０１０１０１０１）$_2$または（１０１０１０１０）$_2$である場合は、Ｈは最小値６４となり、ＤＰＡを用いて鍵を求めるのに必要な計算量は最大でも２６８である。

【０１０３】以上のように、$c_{0,j}$、$c_{1,j}$、・・・、$c_{q-1,j}$に関して条件式（１４）または式（１５）を満たし、かつ$d_{0,j}$、$d_{1,j}$、・・・、$d_{q-1,j}$について式（１６）の条件を満たすことによって、本発明による固定マスク値法を適用した暗号化装置に対してＤＰＡを用いて１２８ビットのラインデール※法の秘密鍵を求める場合、表１および表２に示すように2^{128}または2^{68}に比例した計算量が必要となる。但し、ＤＰＡに対する計算量についての安全性の閾値は2^{64}である。なお、計算量2^{68}は鍵の全てのビット・パターンを計算する場合の計算量2^{128}と比較すると小さいが、そのような鍵は実際の制限された時間内で解読することは実際上不可能である。従って、本発明による固定マスク値法を適用した暗号プロセッサに対してＤＰＡを用いた場合も秘密鍵の解読は実際上不可能である。

【０１０４】

【表１】

表１．固定マスク値法による、Ｓｂｏｘ出力値ロードに対するＤＰＡから１２８ビット秘密鍵を求めるのに必要な計算量とマスク値の関係

$WD_j=(d_{0,j} \oplus d_{1,j}) \vee (d_{1,j} \oplus d_{2,j}) \vee \ldots \vee (d_{q-2,j} \oplus d_{q-1,j})$の値	鍵の解読のための計算量 ($q \geq 2$，任意モデル)
全ての$0 \leq j \leq 15$に関して $WD_j=(11111111)_2$　(固定マスク値法)	2^{128}
一般の場合	2^{128-8F} $F=f_0+f_1+\ldots+f_{15}$, $f_j=0$ (if $WD_j=(11111111)_2$) $f_j=1$ (otherwise)

【０１０５】

【表２】

表２．固定マスク値法による、ＳｕｂｂｙｔｅにおけるＳｂｏｘが１つの場合、Ｓｂｏｘ入力値ロードに対するＤＰＡから１２８ビット秘密鍵を求めるのに必要な計算量とマスク値の関係

$WC_j=(c_{0,j}\oplus c_{1,j})\vee(c_{1,j}\oplus c_{2,j})\vee\ldots\vee(c_{q-2,j}\oplus c_{q-1,j})$の値	鍵の解読のための計算量 (q≥2, 任意モデル)	鍵の解読のための計算量 (q=2, 隣接ビットモデル)
全ての$0\le j\le 15$に関して $(11111111)_2$ (固定マスク値法)	2^{128}	2^{23}
全ての$0\le j\le 15$に関して $(01010101)_2$, $(10101010)_2$ (固定マスク値法)	2^{68}	2^{68}
$WC_j=(wc_{j,7}wc_{j,6}wc_{j,5}wc_{j,4}wc_{j,3}wc_{j,2}wc_{j,1}wc_{j,0})_2$ (一般の場合)	$2^{128-(15/16)H}$ $H=h_0+h_1+\ldots+h_{15}$ $h_j=(wc_{j,e}=0\text{の数})$ $(e=0,1,\ldots,7)$	$2^{128-(15/16)H}$ $H=h_0+h_1+\ldots+h_{15}$ $h=(wc_{j,e}=0\text{の数})+(wc_{j,e}=wc_{j+1,e}=1\text{の数})$ $(e=0,1,\ldots,7)$
全ての$0\le j\le 15$に関して $(00000000)_2$ (安全性が最低の場合)	2^8	2^8

【０１０６】従って、固定マスク値法の安全性は次の通りである。

１．２以上のｑについて、固定マスク値法は、Ｓｂｏｘ出力値のロードに対するＤＰＡに対して、式（１６）の条件を満たせば、鍵の推定に計算量２１２８が必要なので、安全である。この計算量は鍵の全ての可能なパターンをチェックする場合の計算量と同じである。

【０１０７】２．２以上のｑについて、式（１３）の条件を満たす場合、固定マスク値法は、Ｓｂｏｘ入力値ロードに対するＤＰＡに対して、隣接ビット・モデルでない任意のモデルについて式（１５）の条件を満たせば、鍵の推定に計算量２１２８が必要なので、安全である。

【０１０８】３．ｑ＝２について、式（１３）の条件を満たす場合、固定マスク値法は、Ｓｂｏｘ入力値ロードに対するＤＰＡに対して、隣接ビットモデルについて条件式（１４）のｃ０，ｊＸＯＲｃ１，ｊ＝（０１０１０１０１）２または（１０１０１０１０）２を満たせば、鍵の推定に計算量２６８が必要なので、有限時間内での推定は実際上不可能であり、従って安全である。

【０１０９】４．ｑ≧３について、必要なＲＯＭ容量は増加するが、固定マスク値法は、ｑ＝２と同じ解析方法が適用できないため、上述の２の場合と同様に式（１５）の条件を満たせば、、隣接ビット・モデルと任意モデル鍵の双方とも鍵の推定に計算量２１２８が必要なので、安全である。

【０１１０】５．上述の３の場合において、条件式（１４）はＳｂｏｘのセットの数を１つに限定している（ｑ＝２なので出力値の数は２個）が、Ｓｂｏｘのセットの数を相異なるｎ個とするとＲＯＭ容量はｎ倍となるが、計算量は$2^{8(n-1)}\times n16$倍必要となる。

【０１１１】本発明による固定マスク値法は、上述のラインデール法のようなＳＰＮ型の暗号以外に、ＤＥＳのようなフェイステル（Feistel）型の暗号にも適用できる。図３２（Ａ）は通常のＤＥＳによる暗号化の構成を示している。図３２（Ｂ）は図３２（Ａ）におけるＦ関数のより詳細な構成を示している。図３２（Ｂ）において、Ｆ関数は、線形変換ＥおよびＰと、非線形変換テーブル$Ｓ_1$～$Ｓ_8$を有する非線形変換$Ｓ_1$～$Ｓ_8$とを有する。

【０１１２】図３３（Ａ）は、図２１の第１のタイプの暗号化装置１００の一例を示しており、図２９の暗号化装置４００に類似した形態で固定マスク値法を図３２のＤＥＳによる暗号化に適用した暗号化装置７００を示している。図３３（Ｂ）は図３３（Ａ）におけるＦ関数のより詳細な構成を示している。但し、図３３（Ａ）において、図２１におけるプロセッサ１５０、メモリ１６０、１６２および１６４は簡単化のために省略されている。

【０１１３】暗号化装置７００は、乱数ｈを発生する乱数発生器７０１と、その乱数ｈ従って固定マスク値$ＦＭｉｎ_h$の中の１つを選択して供給する選択器７０２と、入力平文と選択された固定マスク値$ＦＭｉｎ_h$とをＸＯＲするＸＯＲ７１２と、入力を受取って乱数ｈおよび拡大鍵Ｋｉに従って入力を暗号化して出力を生成する複数

（例えば１６段）のＦ関数暗号化処理段７１０～７２０と、その乱数ｈ従って固定マスク値ＦＭｏｕｔｈの中の１つを選択して供給する選択器７０４と、Ｆ関数暗号化処理段７２０の出力と選択された固定マスク値ＦＭｏｕｔｈとをＸＯＲして暗号文を生成するＸＯＲ７１４と、を含んでいる。Ｆ関数暗号化処理段７１０～７２０の各々は、前段のＸＯＲ出力を受け取って図３３（Ｂ）に示されたＦ関数を実行し、その出力と前段の出力をＸＯＲ（７２２～７２３）によってＸＯＲして出力を供給する。

【０１１４】図３３（Ｂ）のＦ関数は、乱数ｈに従って選択された固定マスク値ＦＭｉ，ｈを供給する選択器７５９と、拡大鍵Ｋｉと固定マスク値ＦＭｉ，ｈをＸＯＲして出力を供給するＸＯＲ７６２、その出力の値と線形変換Ｅによって線形変換された入力Ｘｉ’とをＸＯＲするＸＯＲ７６３と、その乱数ｈに従ってサブバイト処理ＳｕｂｂｙｔｅＳｉ，ｈ中の１つを選択してＸＯＲ７６３の出力を供給する選択器７５２および７５６と、非線形テーブルＳｂｏｘＳｉ，ｈに従ってサブバイト処理するＳｕｂｂｙｔｅＳｉ，ｈと、その乱数ｈに従ってサブバイト処理Ｓｉ，ｈ中の１つを選択してその出力を供給する選択器７５４および７５７と、選択器７５４および７５７の出力を線形変換して出力Ｚｉ’を供給する線形変換Ｐとを含んでいる。

【０１１５】図２１におけるプロセッサ１５０は、メモリ１６０に格納されているプログラムに従って図３３の暗号化装置７００の諸処理要素７０１～７６３等を制御する。代替構成として、プロセッサ１５０は、諸処理要素７０１～７６３等に対応する機能を実装したメモリ１６０中のプログラムを実行することによって諸処理要素７０１～７６３等を実現してもよい。この場合、図３３はフロー図として見ることができる。

【０１１６】図３４（Ａ）は、図２４の第２のタイプの暗号化装置２００の一例を示しており、図３１の暗号化装置５００に類似した形態で固定マスク値法を図３２のＤＥＳによる暗号化に適用した暗号化装置８００を示している。図３４（Ｂ）は図３４（Ａ）におけるＦ関数のより詳細な構成を示している。但し、図３４（Ａ）において、図２４におけるプロセッサ２５０、メモリ２６０、２６２および２６４は簡単化のために省略されている。

【０１１７】図３４（Ａ）において、暗号化装置８００は、乱数ｈを発生する乱数発生器８０１と、乱数ｈに従って切換えを行う切換器８０２および８０４と、切換器８０２および８０４によって乱数ｈに従って選択される複数の暗号処理部８２０～８３０と、を含んでいる。

【０１１８】図２４におけるプロセッサ２５０は、メモリ２６０に格納されているプログラムに従って図３４の暗号化装置８００の諸処理要素８０１～８６２等を制御する。代替構成として、プロセッサ１５０は、諸処理要素８０１～８６２等に対応する機能を実装したメモリ１６０中のプログラムを実行することによって諸処理要素８０１～８６２等を実現してもよい。この場合、図３４はフロー図として見ることができる。

【０１１９】暗号処理部８２０～８３０の各々は、入力を受け取って出力を生成する複数段（例えば１６段）のＦ関数暗号化処理段８４０～８５０を含んでいる。Ｆ関数暗号化処理段８４０～８５０の各々は、前段のＸＯＲ出力を受け取ってＫｉＸＯＲＦＭｉ，ｈに従って図３４（Ｂ）に示されたＦ関数を実行し、そのＦ関数の出力と前段の出力をＸＯＲ（８２２～８２３）によってＸＯＲして出力を供給する。

【０１２０】図３４（Ｂ）のＦ関数は、拡大鍵Ｋｉと固定マスク値ＦＭｉ，ｈのＸＯＲ値と線形変換Ｅによって線形変換された入力Ｘｉ’とをＸＯＲするＸＯＲ８６２と、非線形テーブルＳｂｏｘＳｉ，ｈによるサブバイト処理ＳｕｂｂｙｔｅＳｉ，ｈ（ｉ＝１，２，・・・，８）と、サブバイト処理Ｓｉ，ｈの出力を線形変換して出力Ｚｉ’を供給する線形変換Ｐとを含んでいる。

【０１２１】図３３および３４において、ラインデール法の場合と同様に入力マスクＦＭｉｎは省略できるが、ラインデール法と同様な手法ではＦＭｏｕｔは省略できない。図３５は、フェイステル型暗号化装置における複数の段の間のマスクの影響の伝播を示している。図３５において太線はマスクされている経路を示している。ＦＭｏｕｔが省略できない理由は、図３５に示されているように、フェイステル型の暗号では或る段でマスクされたデータ（Ａ）が、次段（Ｂ）だけでなくそれ以降の段（Ｃ）へも影響するからである。

【０１２２】従って、フェイステル型暗号では、前段のマスク値を次段で完全にキャンセルできない。図３６は、フェイステル型暗号化装置におけるマスク発生からマスク値キャンセルまでの経路を示している。図３６において、太線はマスクされている経路を示している。フェイステル型暗号ではマスクを行ってからマスク値をキャンセルするまでには図３６に示されているように４段以上必要であり、４段以上にわたってマスク値をキャンセルできる。このマスク値キャンセルの手法は、ＳＰＮ型暗号化におけるように非線形変換の出力マスクが任意選択可能であることを利用するのでなくて、図３６に示されているように前段からのマスクと等しい出力マスクを生成して、フェイステル型暗号化におけるＸＯＲによってキャンセルするものである。

【０１２３】この方法を用いれば暗号化処理最後のマスクＦＭｏｕｔが不要になる。この手法をＤＥＳ等の４ラウンド以上の暗号に適用する場合は、４ラウンドでキャンセルするような固定マスク値の構成を繰り返してもよいし、または全ラウンドの最後でキャンセルするような固定マスク値を用いてもよい。

【０１２４】上述の実施形態においては、全てのラウン

31

ドに対して固定マスク値法を適用する場合について説明した。しかし、ＤＰＡが可能となる条件で述べたように、ＤＰＡが成功するためには入力値が既知でかつ攻撃者がその値を制御可能でなければならない。従って、暗号化処理の最初の数ラウンドに対して固定値マスク法を適用すれば、その後のラウンドはその入力が未知でかつ制御不可能なのでＤＰＡ対策は不要となる。それによって暗号化対策に必要な処理を削減できる。

【０１２５】表３は、ラインデール法の実装における固*

32

*定マスク値法を用いた暗号化の結果と、ＤＰＡ対策なしの通常の暗号化、および従来の乱数マスク値法を用いた暗号化との比較を示している。表３において、Ｓは対策なしの場合の処理時間を表し、Ｒは対策なしの場合の必要なＲＡＭ量を表し、Ｍは対策なしの場合の必要なＲＯＭ量を表す。但し、Ｒ≪Ｍである。安全性は攻撃者が鍵を推定するのに必要なチェックすべき可能な鍵の数で表される。

【表３】

表 3. ラインデール法における、固定マスク値法適用、ＤＰＡ対策なしの場合、および乱数マスク値法適用の比較　（但しＲ≪Ｍ）

	対策なし	固定マスク値法		乱数マスク値法
		q≥２ 任意モデル	q＝２ 隣接モデル	
処理時間	S	≒S	≒S	>>S (3S~5S)
ＲＡＭ量	R	≒R	≒R	≧R+M
ＲＯＭ量	M	qM	2M	≒M
安全性	1	$\geq 2^{128}$	2^{72}	$>> 2^{128}$

【０１２６】表４は、ＤＥＳの実装における固定マスク値法を用いた暗号化の結果と、ＤＰＡ対策なしの通常の暗号化、および従来の乱数マスク値法を用いた暗号化と※

※の比較を示している。

【表４】

表４．ＤＥＳにおける、固定マスク値法適用、ＤＰＡ対策なしの場合、および乱数マスク値法適用の比較　（但しＲ≪Ｍ）

	対策なし	固定マスク値法	乱数マスク値法
		q＝２	
処理時間	S	≒S	>>S (3S~5S)
ＲＡＭ量	R	≒R	≧R+M
ＲＯＭ量	8M	16M	≒8M
安全性	1	$>> 2^{56}$	$>> 2^{56}$

【０１２７】表３および表４より、乱数マスク値法では処理時間が長く大きいＲＡＭ量を必要とするのに対して、固定マスク値法では、２～３倍のＲＯＭ量を必要とするが大きいＲＡＭ領域を必要とせずに、対策なしの場合とほぼ同じ程度の処理時間で充分な安全性が確保できる。

【０１２８】上述の実施形態では主にラインデール法およびＤＥＳについて説明したが、固定マスク値法は、ラインデール法以外のＳＰＮ型暗号化、ＤＥＳ以外のフェイステル型暗号化、およびそれを組合せた型の暗号にも適用でき同様の有効性を示す。

【０１２９】以上説明した実施形態は典型例として挙げたに過ぎず、その変形およびバリエーションは当業者にとって明らかであり、当業者であれば本発明の原理および請求の範囲に記載した発明の範囲を逸脱することなく上述の実施形態の種々の変形を行えることは明らかである。

【０１３０】（付記１）　排他的論理和手段および非線形変換手段を有する暗号化装置であって、乱数を発生する乱数発生器手段と、ｑ個の固定値（ここでｑは整数である）と、前記乱数に従って前記ｑ個の固定値の中の１つを選択する第１の選択器と、を具え、前記排他的論理和手段は、前記選択された固定値と鍵の排他的論理和と前記排他的論理和手段の入力の排他的論理和をとるものである、暗号化装置。

（付記２）　さらに、ｑ組（セット）のマスクされた固

定テーブルと、前記乱数に従って前記ｑ組の固定テーブルの中の１組を選択する第２の選択器と、を具え、前記非線形変換手段は前記選択された１組の固定テーブルに従って入力を非線形変換するものである、付記１に記載の暗号化装置。

（付記３）　前記第１の排他的論理和手段および前記非線形変換手段を有する暗号化部と、前記暗号化装置への入力と前記乱数に従って選択された或る固定値の排他的論理和をとる第２の排他的論理和手段と、前記乱数に従って選択された或る固定値と前記暗号化部の出力の排他的論理和をとる第３の排他的論理和手段とを具える、付記１に記載の暗号化装置。

（付記４）　排他的論理和手段および非線形変換手段を有する暗号化装置であって、乱数を発生する乱数発生器手段と、ｑ組のマスクされた固定テーブル（ここでｑは整数である）と、前記乱数に従って前記ｑ組の固定テーブルの中の１組を選択する選択器と、を具え、前記非線形変換手段は、前記選択された１組の固定テーブルに従って入力を非線形変換するものである、暗号化装置。

（付記５）　複数の暗号化段を具え、この複数の暗号化段の各々は、その段の、前記排他的論理和手段と、前記固定テーブルと、前記選択器と、を有するものであり、前記複数の暗号化段のそれぞれの前記固定テーブルは同じものである、付記４に記載の暗号化装置。

（付記６）　マスクされる前の固定テーブルをＳ［ｘ］とし、マスクされたｊ番目のテーブルを$S_j[x \mathrm{XOR} c_{i,j}] \mathrm{XOR} d_{i,j}$（ｊ＝０，１，・・・，１５）としたときに、$(c_{0,j} \mathrm{XOR} c_{1,j}) \vee \mathrm{XOR} (c_{1,j} \mathrm{XOR} c_{2,j}) \vee \cdots \vee (c_{q-2,j} \mathrm{XOR} c_{q-2,j}) = (11111111)_2$である、付記４に記載の暗号化装置。

（付記７）　テーブルの組の数ｑ＝２とする場合に、マスクされる前の固定テーブルをＳ［ｘ］とし、マスクされたｊ番目のテーブルを$S_j[x \mathrm{XOR} c_{i,j}] \mathrm{XOR} d_{i,j}$（ｊ＝０，１，・・・，１５）としたときに、$c_{0,j} \mathrm{XOR} c_{1,j} = (10110101)_2$または$(01101010)_2$である、付記４に記載の暗号化装置。

（付記８）　マスクされる前の固定テーブルをＳ［ｘ］とし、マスクされたｊ番目のテーブルを$S_j[x \mathrm{XOR} c_{i,j}] \mathrm{XOR} d_{i,j}$（ｊ＝０，１，・・・，１５）としたときに、$(d_{0,j} \mathrm{XOR} d_{1,j}) \vee (d_{1,j} \mathrm{XOR} d_{2,j}) \vee \cdots \vee (d_{q-2,j} \mathrm{XOR} d_{q-2,j}) = (11111111)_2$である、付記４に記載の暗号化装置。

（付記９）　前記非線形変換手段はサブバイト手段であり、さらに、入力をシフトする手段と、入力をミクストコラムする手段とを具える、付記４に記載の暗号化装置。

（付記１０）　乱数を発生する乱数発生器手段と、並列

に結合された複数の暗号化部と、前記乱数に従って前記複数の暗号化部の中の１つを選択する選択器と、を具え、前記複数の暗号化部の各々は、排他的論理和手段および非線形変換手段を含むものである、暗号化装置。

（付記１１）　前記選択された暗号化部の前記排他的論理和手段は、固定値と鍵の排他的論理和と前記排他的論理和手段の入力の排他的論理和をとるものである、付記１０に記載の暗号化装置。

（付記１２）　前記非線形変換手段は、固定テーブルに従って前記非線形変換手段の入力を非線形変換するものである、付記１０に記載の暗号化装置。

（付記１３）　前記複数の暗号化部の各々は、その暗号化部への入力と固定値の排他的論理和をとる第２の排他的論理和手段と、入力と固定値の排他的論理和をその暗号化部の出力として生成する第３の排他的論理和手段と、を具えるものである、付記１０に記載の暗号化装置。

（付記１４）　前記選択された暗号化部の前記非線形変換手段は、固定テーブルに従って前記非線形変換手段の入力を非線形変換するものである、付記１０に記載の暗号化装置。

（付記１５）　前記複数の暗号化部の各々は複数の暗号化段を含んでおり、前記複数の暗号化段の各々は、その段の、鍵と固定値の排他的論理和と入力の排他的論理和をとる排他的論理和手段と、固定テーブルに従って入力を非線形変換する非線形変換手段とを含むものである、付記１０に記載の暗号化装置。

（付記１６）　乱数を発生する乱数発生器手段と複数の暗号化段とを具えた暗号化装置であって、前記複数の暗号化段の各々は、入力を非線形変換する非線形変換手段と、第１の入力と第２の入力の排他的論理和をとる排他的論理和手段と、を含み、前記排他的論理和手段の第２の入力は前記非線形変換手段の出力に結合されており、前記非線形変換手段は、ｑ個の固定値（ここでｑは整数である）と、前記乱数に従って前記ｑ個の固定値の中の１つを選択する選択器と、前記選択された固定値と鍵の排他的論理和と入力の排他的論理和をとる別の排他的論理和手段と、を有するものである、暗号化装置。

（付記１７）　前記非線形変換手段は、さらに、前記非線形変換手段の内部に、固定テーブルに従って入力を非線形変換する複数の非線形変換手段と、前記複数の非線形変換手段の中の１つのを選択する選択器とを有するものである、付記１６に記載の暗号化装置。

（付記１８）　前記複数の暗号化段のそれぞれの前記非線形変換手段の固定テーブルは同じである、付記１７に記載の暗号化装置。

（付記１９）　前記複数の暗号化段における連続する複数の段においてマスクをキャンセルする、付記１６に記載の暗号化装置。

（付記２０）　前記複数の暗号化段において前記複数の

暗号化段の数より少ない数の暗号化段においてマスクが行われる、付記１６に記載の暗号化装置。

（付記２１）　乱数を発生する乱数発生器手段と複数の暗号化段とを具えた暗号化装置であって、前記複数の暗号化段の各々は、入力を非線形変換する非線形変換手段と、第１の入力と第２の入力の排他的論理和をとる排他的論理和手段と、を含み、前記排他的論理和手段の第２の入力は前記非線形変換手段の出力に結合されており、前記非線形変換手段は、内部に、前記乱数に従って、固定テーブルに従って入力を非線形変換する非線形変換手段を有するものである、暗号化装置。

（付記２２）　乱数を発生する乱数発生器手段と、並列に結合された複数の暗号化部と、前記乱数に従って前記複数の暗号化部の中の１つを選択する選択器と、を具え、前記複数の暗号化部の各々は複数の暗号化段を有し、前記複数の暗号化段の各々は、入力を非線形変換する非線形変換手段と、第１の入力と第２の入力の排他的論理和をとる排他的論理和手段と、を含み、前記排他的論理和手段の第２の入力は前記非線形変換手段の出力に結合されているものである、暗号化装置。

（付記２３）　前記非線形変換手段は、固定値と鍵の排他的論理和と入力の排他的論理和をとる別の排他的論理和手段と、前記非線形変換手段の内部に、固定テーブルに従って入力を非線形変換する非線形変換手段と、を有するものである、付記２２に記載の暗号化装置。

（付記２４）　マスクされる前の固定テーブルを$S_j[x]$とし、マスクされたｊ番目のテーブルを$S_j'[x \mathrm{XOR} c_{i,j}] \mathrm{XOR} d_{i,j}$（ｊ＝０，１，・・・，７）としたときに、$(d_{0,j} \mathrm{XOR} d_{1,j}) \vee (d_{1,j} \mathrm{XOR} d_{2,j}) \vee \cdots \vee (d_{q-2,j} \mathrm{XOR} d_{q-2,j}) = (1111)_2$である、付記２２に記載の暗号化装置。

（付記２５）　暗号化装置において用いるための記憶媒体に格納されたプログラムであって、乱数に従ってｑ個の固定値の中の１つを選択するステップ（ここでｑは整数である）と、前記選択された固定値と鍵の排他的論理和と入力値の排他的論理和をとるステップと、前記乱数に従ってｑ組のマスクされた固定テーブルの中の１組を選択するステップと、前記選択された１組の固定テーブルに従って入力値を非線形変換するステップと、を実行させる、プログラム。

（付記２６）　暗号化装置において用いるための記憶媒体に格納されたプログラムであって、乱数に従って複数の暗号化手順の中の１つを選択するステップと、前記選択された暗号化手順に従って、入力値を暗号化して出力するステップと、を実行させ、前記暗号化するステップは、固定値と鍵の排他的論理和と入力値の排他的論理和をとるステップと、１組の固定テーブルに従って入力値を非線形変換するステップと、を含むものである、プログラム。

（付記２７）　暗号化装置において用いるための記憶媒体に格納されたプログラムであって、入力値を非線形変換して出力を供給するステップと、第１の入力値と前記出力である第２の入力値の排他的論理和をとるステップと、を実行させ、前記非線形変換するステップは、乱数に従ってｑ個の固定値の中の１つを選択するステップ（ここでｑは整数である）と、前記選択された固定値と鍵の排他的論理和と入力値の排他的論理和をとるステップと、前記乱数に関連付けられた固定テーブルに従って入力値を非線形変換するステップと、を含むものである、プログラム。

（付記２８）　暗号化装置において用いるための記憶媒体に格納されたプログラムであって、乱数に従って複数の暗号化手順の中の１つを選択するステップと、前記選択された暗号化手順に従って、入力値を暗号化して出力を供給するステップと、を実行させ、前記暗号化するステップは、入力値を非線形変換して出力を供給するステップと、第１の入力値と前記出力である第２の入力値の排他的論理和をとるステップと、を含むものである、プログラム。

【０１３１】

【発明の効果】本発明は、上述の特徴によって、共通鍵暗号を行う暗号プロセッサに対する効率的な暗号解読防止を実現でき、秘密鍵の推定を困難にし、暗号プロセッサの安全性を高めることができるという効果を奏する。

【図面の簡単な説明】

【図１】図１は、図１はスマートカードにおける共通秘密鍵を用いた暗号化の例を示している。

【図２】図２は、典型的な共通鍵暗号で用いられる鍵ＸＯＲを示している。

【図３】図３は、典型的な共通鍵暗号で用いられる線形変換を示している。

【図４】図４は、典型的な共通鍵暗号で用いられる非線形変換を示している。

【図５】図５は、縦続接続の形の鍵ＸＯＲ演算（図２）と非線形変換（図４）の組合せである暗号化の例を示している。

【図６】図６は、図５における任意の非線形変換要素ｗｉに関係する構成要素を示している。

【図７】図７（Ａ）および（Ｂ）は、暗号化プロセッサがその入力平文に応答して消費する電力消費の時間変化を表す電力消費曲線を示している。図７（Ｃ）はスパイクを有する電力消費曲線の差分を示している。図７（Ｄ）はスパイクのない電力消費曲線の差分を示している。

【図８】図８は、図４の暗号化器の前後に２つの線形変換を追加した構成を有する暗号化器を示している。

【図９】図９は、図４の暗号化器における電力消費曲線の測定対象点Ａ、ＢおよびＣを示している。

【図１０】図１０は、乱数マスク値法による処理の概要

37

図を示している。

【図１１】図１１は、乱数マスク値法による鍵ＸＯＲを示している。

【図１２】図１２は、乱数マスク値法による線形関数を示している。

【図１３】図１３は、乱数マスク値法による非線形関数を示している。

【図１４】図１４は、ＤＰＡ対策のない通常のＮ段ラインデール処理の全体的構成を示している。

【図１５】図１５は、ラインデール法の秘密鍵Ｋｓｅｃから拡大鍵K_0、K_1、・・・、K_Nを生成する拡大鍵生成器を示している。

【図１６】図１６は、サブバイト処理Ｓｕｂｂｙｔｅの構成を示している。

【図１７】図１７は、シフトＳｈｉｆｔの構成を示している。

【図１８】図１８は、ミクストコラム処理Ｍｉｘｅｄｃｏｌｕｍｎの構成を示している。

【図１９】図１９は、図１４の通常のＮ段ラインデール法に対して、乱数マスク値法を適用したＮ段ラインデール法を示している。

【図２０】図２０は、図１９の１６個のＳｂｏｘで使用されるＮｅｗＳｂｏｘを示している。

【図２１】図２１は、本発明による第１のタイプの暗号化装置の概略的構成を示している。

【図２２】図２２は、図２１における鍵ＸＯＲの構成を示している。

【図２３】図２３は、図２１における非線形変換の構成を示している。

【図２４】図２４は、本発明による第２のタイプの暗号化装置の概略的構成を示している。

【図２５】図２５は、図２４における鍵ＸＯＲの構成を示している。

【図２６】図２６は、図２４における非線形変換の構成を示している。

【図２７】図２７は、図２１の第１のタイプの暗号化装置の一例を示している。

【図２８】図２８は、図２７におけるＳｕｂｂｙｔｅの構成を示している。

38

*【図２９】図２９は、図２１の第１のタイプの暗号化装置の別の例を示している。

【図３０】図３０は、図２９におけるＳｕｂｂｙｔｅの構成を示している。

【図３１】図３１は、図２４の第２のタイプの暗号化装置の一例を示している。

【図３２】図３２は、通常のＤＥＳの構成を示している。

【図３３】図３３は、図２９の固定マスク値法をフェイステル型のＤＥＳに適用した構成を示している。

【図３４】図３４は、図３１の固定マスク値法をフェイステル型のＤＥＳに適用した構成を示している。

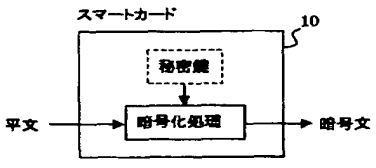
【図３５】図３５は、フェイステル型暗号化装置における暗号における段の間のマスクの伝播を示している。

【図３６】図３６は、フェイステル型暗号化装置におけるマスク発生からマスク値キャンセルまでの経路を示している。

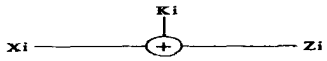
【符号の説明】

１００　暗号化措置
１０１　暗号化処理部
１０３　乱数発生器
１０４、１０５　切換器
１０６、１０７　論理ＸＯＲ
２００　暗号化装置
２０３　乱数発生器
２０８～２０９　暗号化処理部
２０４、２０５　論理ＸＯＲ
２１１、２１３　切換器
３００　暗号化装置
３１０　第ｉ段の処理
３１１　第Ｎ－１段の処理
３０３　乱数発生器
３０５、３２９、３３９、３７９、３９９　切換器
３０２、３３１、３３３、３７１、３７３、３８３　ＯＸＲ（排他的論理和）
３３４　Ｓｕｂｂｙｔｅ
３３５　Ｓｈｉｆｔ
３３６　Ｍｉｘｅｄｃｏｌｕｍｎ

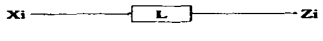
【図１】



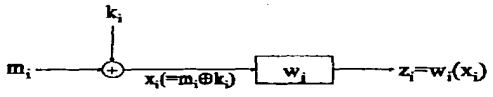
【図２】



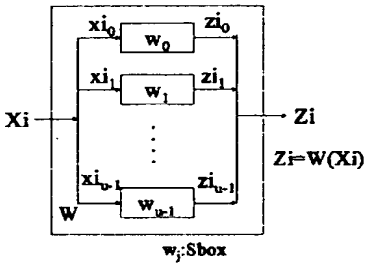
【図３】



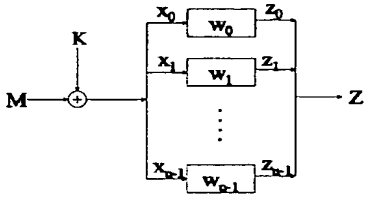
【図６】



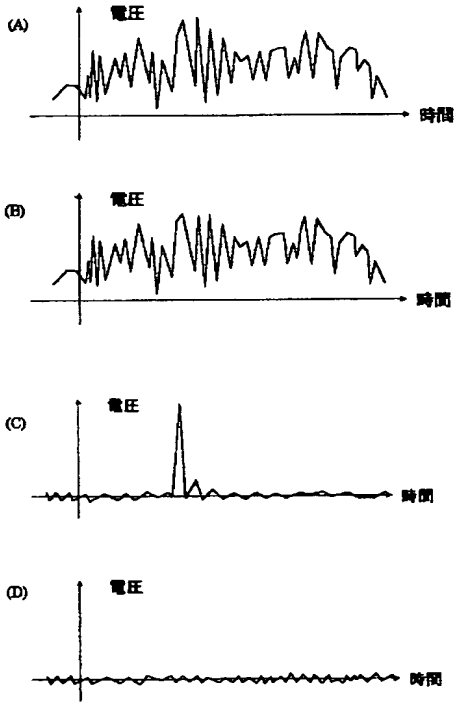
【図４】



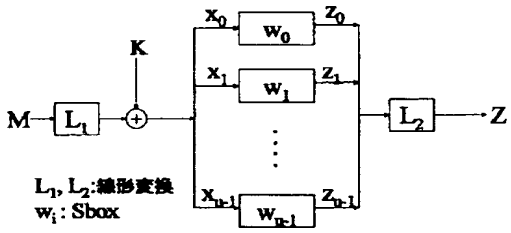
【図５】



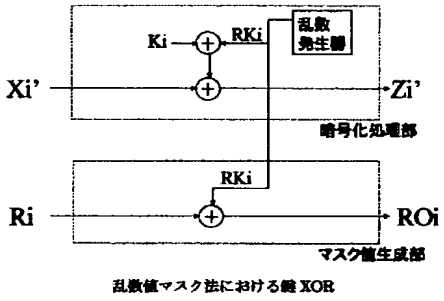
【図７】



【図８】

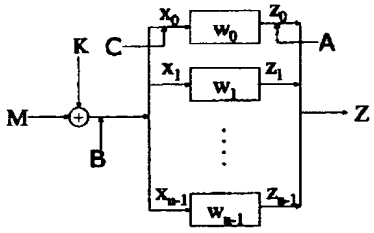


【図１１】

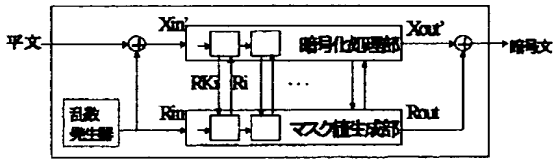


乱数値マスク法における鍵 XOR

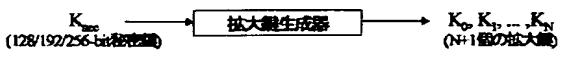
【図９】



【図１０】

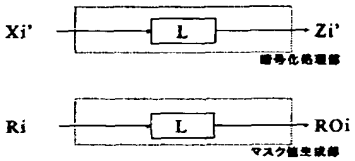


【図１５】

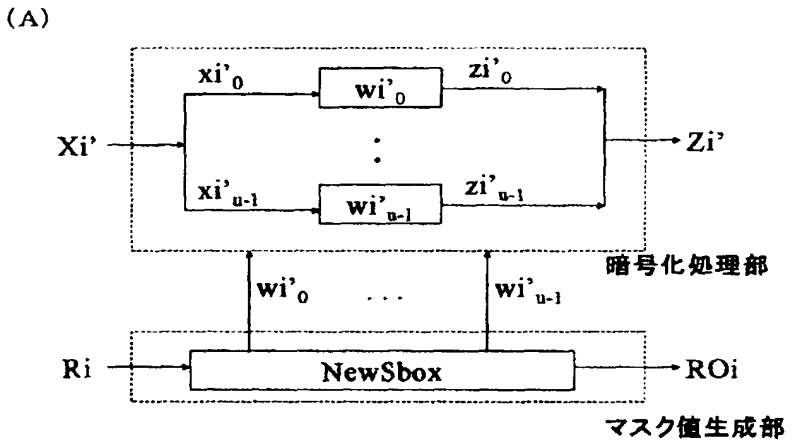


Rijndael における拡大鍵の生成

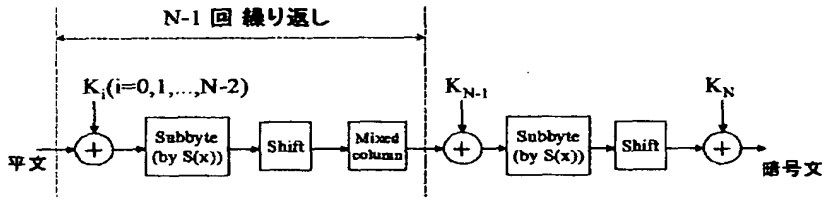
【図１２】



【図１３】

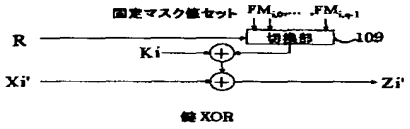


【図１４】

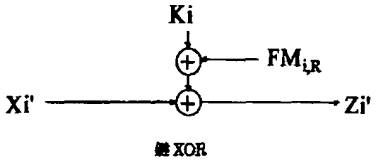


通常の Rijndael 処理

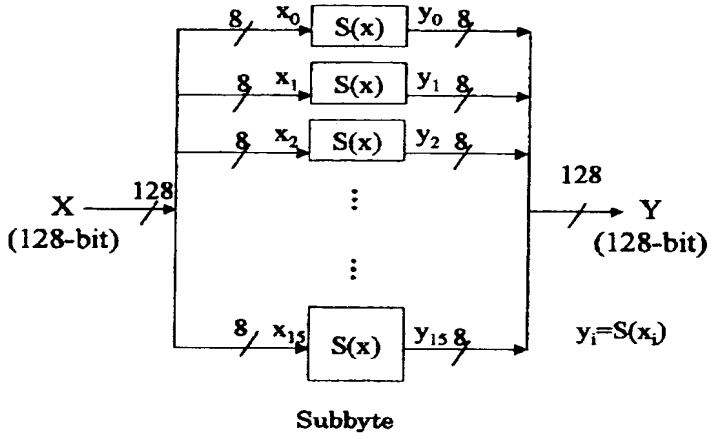
【図２２】



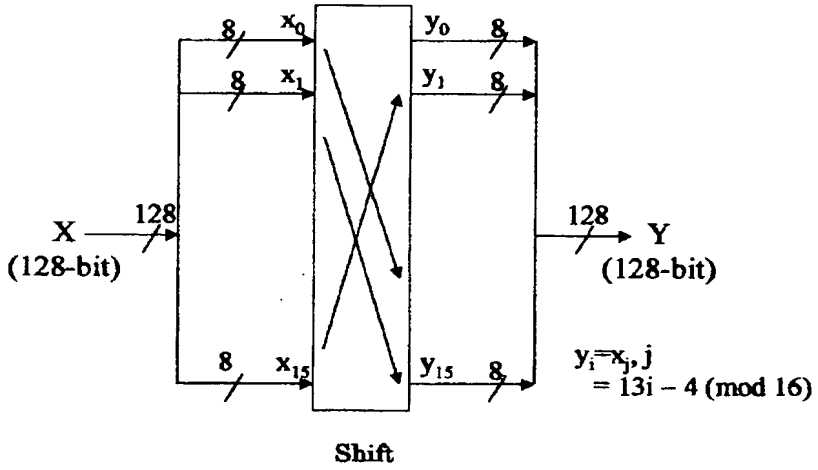
【図２５】



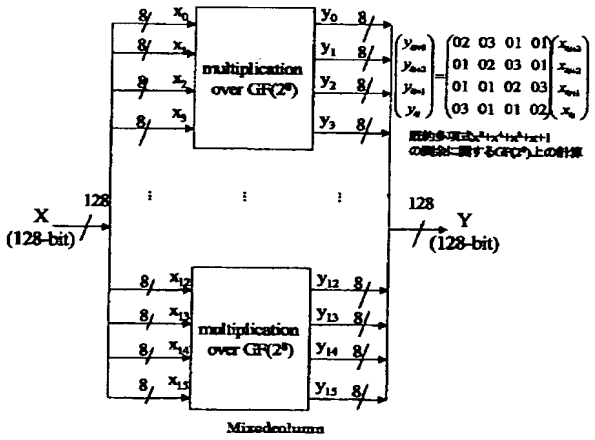
【図１６】



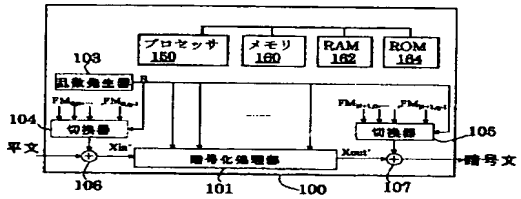
【図１７】



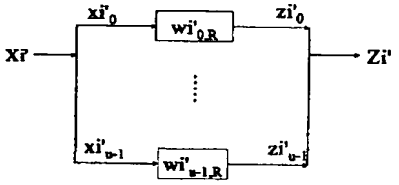
【図１８】



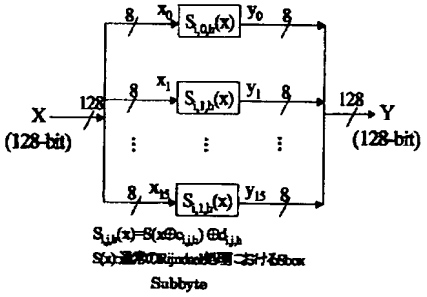
【図２１】



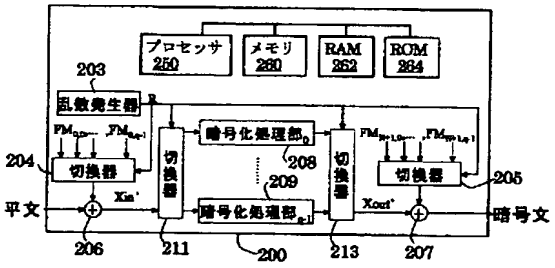
【図２６】



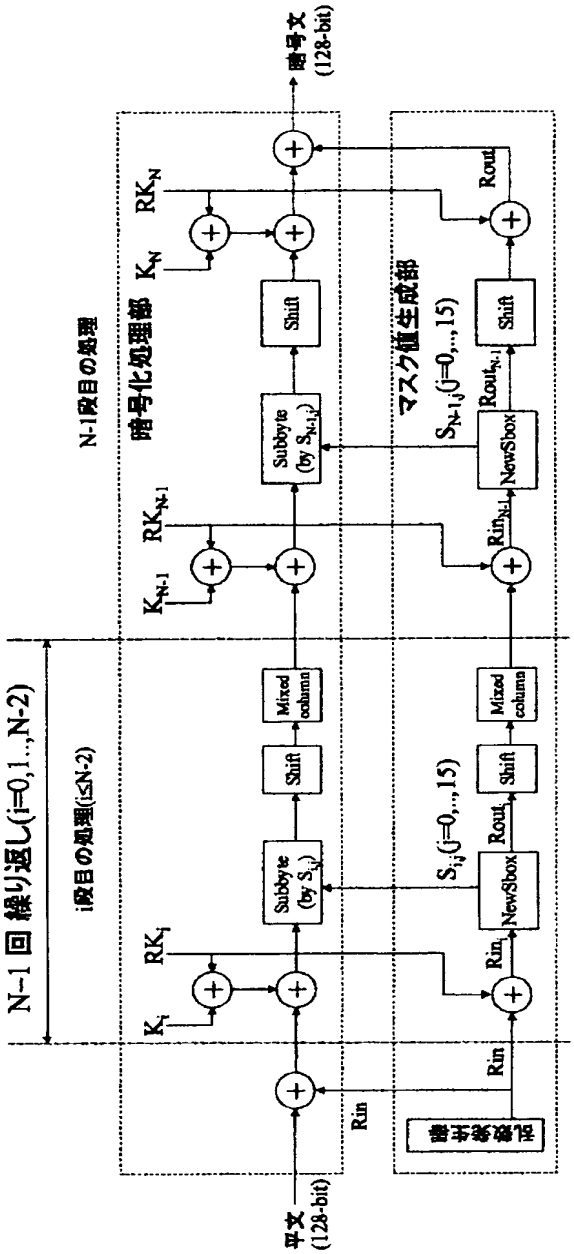
【図２８】



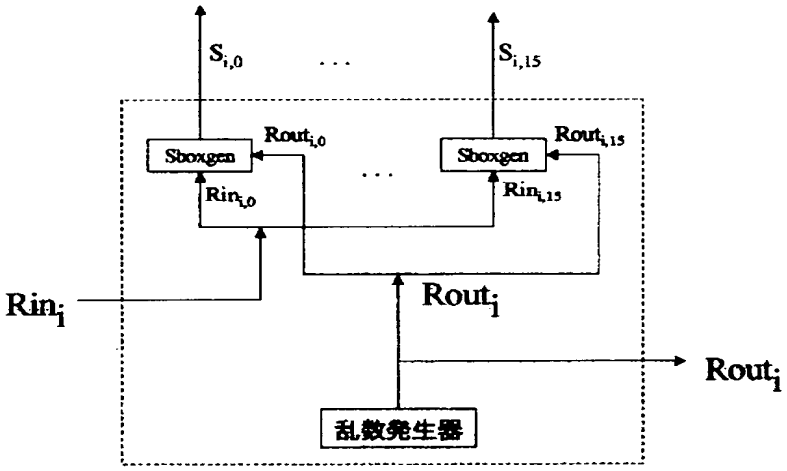
【図２４】



【図１９】


暗号文
(128-bit)
N-1段目の処理
暗号化処理部
マスク値生成部
Shift
Subbyte
(by $S_{N-1,j}$)
NewSbox
Mixed column
Rout
Rin
RK_N
K_N
RK_{N-1}
K_{N-1}
RK_i
K_i
$S_{N-1,j}$(j=0,..,15)
$S_{i,j}$(j=0,..,15)
$Rout_{N-1}$
$Rout_i$
Rin_{N-1}
Rin_i
N-1 回 繰り返し(i=0,1..,N-2)
i段目の処理(i≦N-2)
平文
(128-bit)

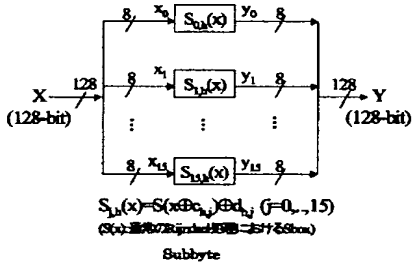
【図２０】



Sboxgen:$S_{i,j}(x)=S(x\oplus Rin_{i,j})\oplus Rout_{i,j}$を満たすSbox, $S_{i,j}$の生成

NewSbox

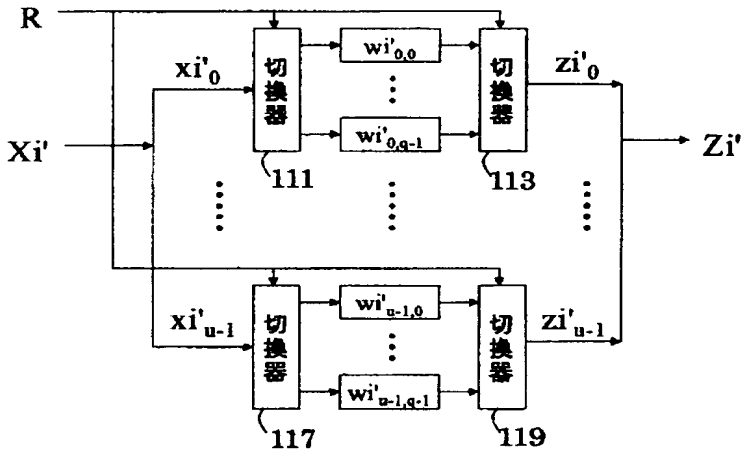
【図３０】



$S_{i,0}(x)=S(x\oplus c_{0,j})\oplus d_{0,j}$ (j=0,...,15)

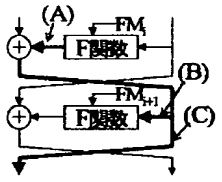
Subbyte

【図２３】



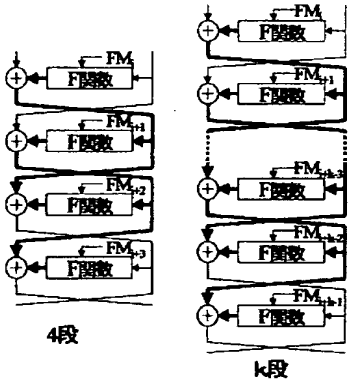
非線形変換

【図３５】



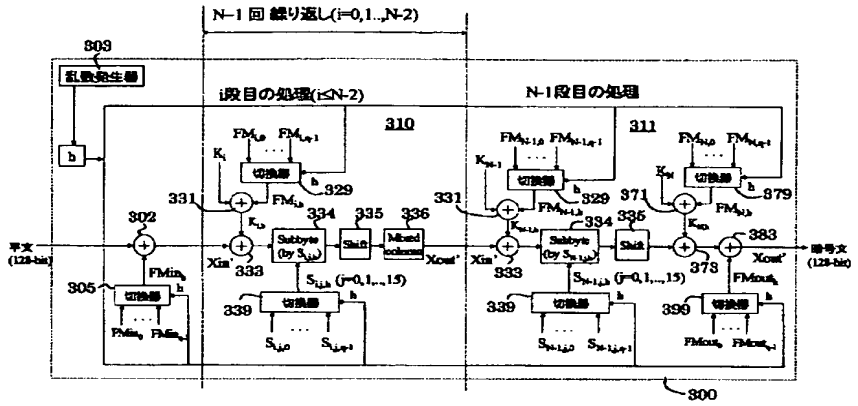
Feistel 型のマスクの伝播

【図３６】

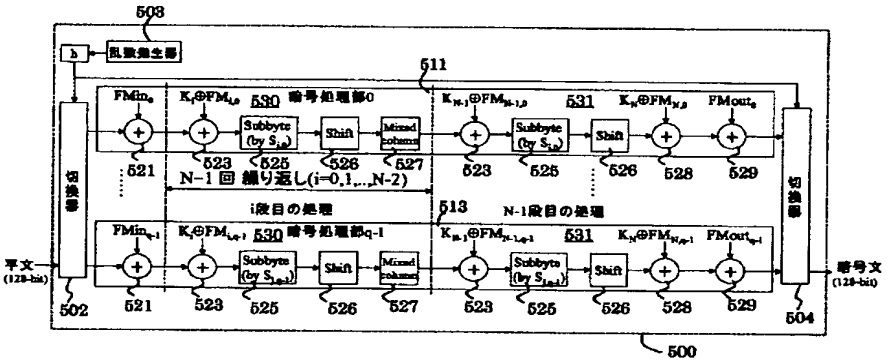


Feistel 型暗号化装置における
マスク値の発生からキャンセルまでの経路

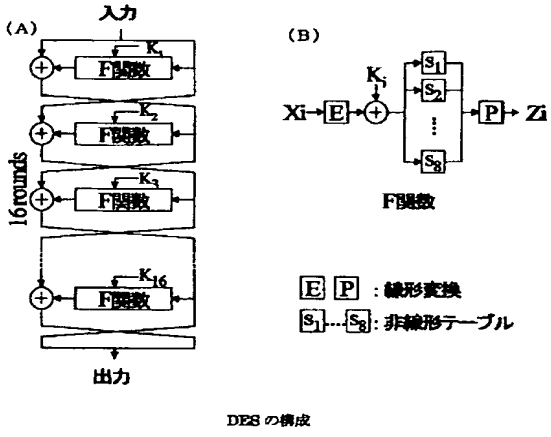
【図２７】



【図３１】

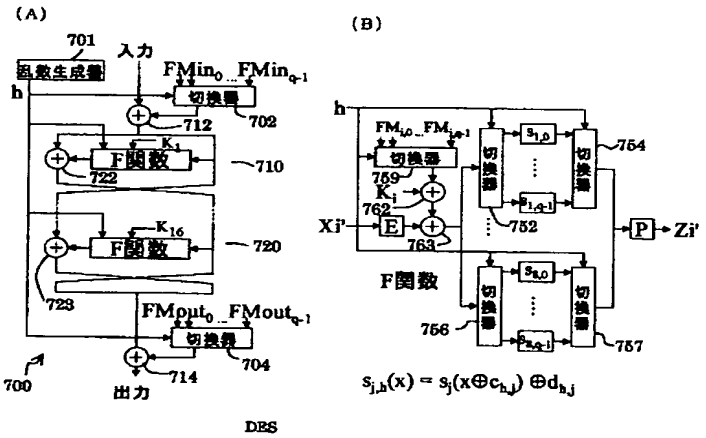


【図３２】



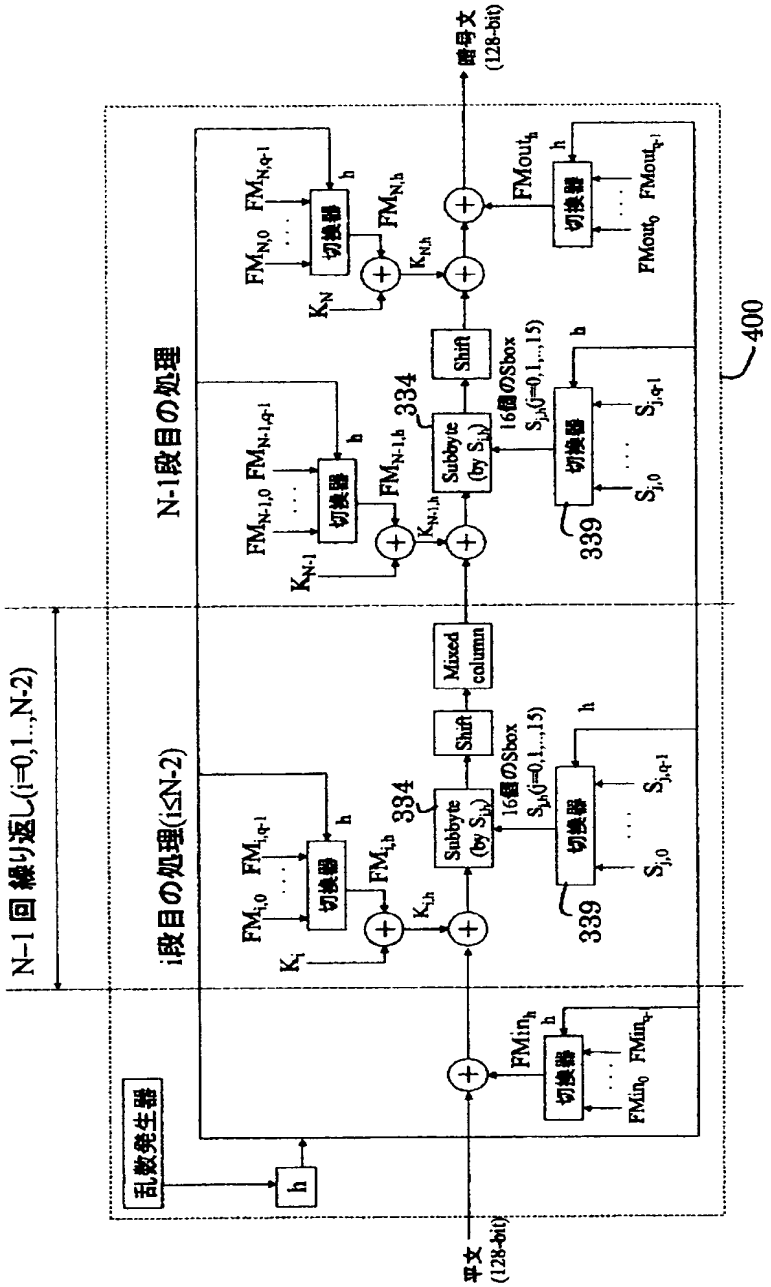
DES の構成

【図３３】

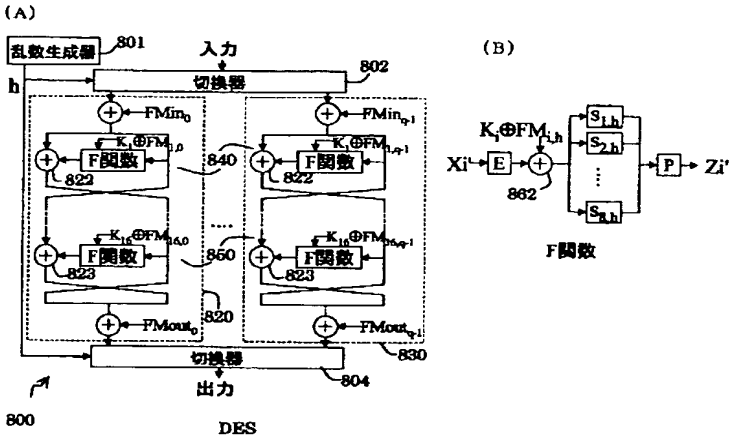


DES

【図２９】


N-1 回 繰り返し(i=0,1..,N-2)
i段目の処理(i≤N-2)
N-1段目の処理
乱数発生器
h
平文 (128-bit)
暗号文 (128-bit)
切換器
Subbyte (by $S_{i,h}$)
16個のSbox $S_{i,h}$(i=0,1,..,15)
Shift
Mixed column
334
339
400
K_i
$K_{i,h}$
K_{N-1}
$K_{N-1,h}$
K_N
$K_{N,h}$
$FM_{i,0}$ $FM_{i,q-1}$ $FM_{i,h}$
$FM_{N-1,0}$ $FM_{N-1,q-1}$ $FM_{N-1,h}$
$FM_{N,0}$ $FM_{N,q-1}$ $FM_{N,h}$
$FMin_0$ $FMin_{q-1}$ $FMin_h$
$FMout_0$ $FMout_{q-1}$ $FMout_h$
$S_{j,0}$ $S_{j,q-1}$

【図３４】



フロントページの続き

(72) 発明者　鳥居　直哉
神奈川県川崎市中原区上小田中４丁目１番
１号　富士通株式会社内

Ｆターム(参考)　5J104 AA41 AA47 FA00 JA03 NA02